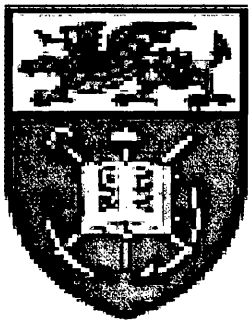
Application of modern control techniques in AC speed drive system.

Jamal, Wissam

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING



UNIVERSITY OF WALES SWANSEA

APPLICATION OF MODERN CONTROL TECHNIQUES IN AC SPEED DRIVE SYSTEM

BY

WISSAM JAMAL, B.Eng. (HONOUR)

Thesis submitted to the University of Wales in candidature for the degree of

Doctor of Philosophy

February, 2002

DECLERATION

This work has not been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... (candidate)
Date.....20/02/2002.....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ..... (candidate)
Date.....20/02/2002.....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)
Date.....20/02/2002.....

I dedicate this work to the memory of
my mother, **Amna Jamal**

ACKNOWLEDGEMENT

I would like to begin my acknowledgement by thanking my director of studies Dr. M. S. Khanniche for his unparalleled guidance and constructive suggestions throughout the period of this research.

I would like also to express my sincere thanks and respect to my father Mohammed Jamal for his financial and continuous moral support. Sincere thanks also goes to my beloved sisters and brothers, Fatima, Magedah, Amal, Adnan, Yahya, and Omar Jamal for their heartfelt encouragement. Special thanks goes to Michelle Jamal for her remarkable help correcting the English of the thesis.

At last but not least, I would like to thank all my colleagues in the Power Electronics group and close friends for their support and encouragement.

SUMMARY

In the past, Direct Current (dc) machines have been commonly favoured in areas where a precise variable speed operation is highly required. This is due to the feasible linear control of flux and torque, which is accomplished by simply varying the field and armature currents. However, they are bulky, expensive and require periodic maintenance due to the existence of commutators and brushes. Alternating Current (ac) machines particularly the squirrel cage induction type have emerged as an alternative to those of dc machines in the application of speed drive systems. In general, however, they do require more complex control schemes than the dc motors, because of their highly non-linear dynamic structure with strong dynamic interactions.

This situation has changed dramatically over the last few years with the advent of fast switching power converters along with high performance micro-controllers, which made a significant contribution to performance enhancement of modern speed drive systems. In addition, various control techniques have made possible the application of induction motors in high performance speed drive operations where traditionally only dc motors were previously available. On the other hand, in many speed drive applications which incorporate either scalar or vector control, the prime objective of the speed controller is the capability of achieving a good speed tracking performance and without sensitivity to parameters and operating condition changes.

For these reasons, comprehensive investigation of state-of-the-art modern control schemes, which include fuzzy logic and sliding mode control are discussed. The main principles underlying fuzzy logic and sliding mode control schemes along with their basic theory and general mathematical representation are reviewed. In addition, the application of fuzzy logic concepts to reduce the chattering phenomenon typically inherited in the sliding mode control is successfully presented, which results in a new integrated fuzzy sliding mode control algorithms.

Through extensive simulation studies, it is found that the fuzzy logic control scheme attained a good transient performance for the speed drive system in comparison to the conventional sliding mode control and the new integrated fuzzy sliding mode control. Furthermore, the design simplicity of the fuzzy logic control system has made it virtually attractive for the ease of practical implementation of the proposed drive system.

Extensive practical testes of the proposed variable speed drive system have been carried out to verify the validity of the simulation analysis of the proposed fuzzy logic control system. Several tests are conducted in order to bring out the effectiveness of the designed control system upon step change in speed command and impact load disturbances. The digital implementation of the proposed fuzzy logic control algorithms is realised on a single chip, Intel 80C196KC 16-bit embedded microcontroller, a low cost derivative of the MCS-96 architecture.

The main contribution of this thesis is the novel approach to design a sliding mode control system using concepts from fuzzy logic algorithms to alleviate the chattering problems and improve the dynamics of the induction motor drive

Contents

Chapter 6:

Control methods of an induction motor

ABBREVIATIONS

ac	:	alternating current
A/D	:	analogue to digital converter
ASM196	:	assembly language compiler
dc	:	direct current
IM	:	induction motor
rms	:	root mean square value
VSI	:	Voltage Source Inverter
CSI	:	Current Source Inverter
PWM	:	Pulse Width Modulation
SPWM	:	Sinusoidal Pulse Width Modulation
SVPWM	:	Space Vector Pulse Width Modulation
HPWM	:	Hysteresis Pulse Width Modulation
PID	:	Proportional + Integral + Derivative control
PI	:	Proportional + Integral control
FOC	:	Field Oriented Control
IOLC	:	Input-Output Linearisation Control
FLC	:	Fuzzy Logic Control
DSP	:	Digital Signal Processor
DSC	:	Direct Self Control
PLL	:	Phase Locked Loop control
VSC-SLM	:	Variable Structure Control with Sliding Mode
FSLMC	:	Fuzzy Sliding Mode Control
SCWRIM	:	Series Connected Wound Rotor Induction Machine
CAC	:	Chattering Alleviation Control
IGBT	:	insulated gate bipolar transistor
RAM	:	random access memory
SFR	:	special function register
HP	:	horse power
UART	:	universal asynchronous receiver and transmitter
EX-OR	:	Exclusive OR logic gate
HSO	:	High Speed Output
HSI	:	High Speed Input
SWT	:	Software Timer
mmf	:	magnetomotive force
emf	:	electromotive force
MAF	:	Moving Average Filter
NB	:	negative big fuzzy set
NM	:	negative medium fuzzy set
NS	:	negative small fuzzy set
ZO	:	near zero fuzzy set
PS	:	positive small fuzzy set
PM	:	positive medium fuzzy set
PB	:	positive big fuzzy set

VL	:	Very Large fuzzy set for sliding surface *s*
L	:	Large fuzzy set for sliding surface *s*
M	:	Medium fuzzy set for sliding surface *s*
S	:	Small fuzzy set for sliding surface *s*
VS	:	Very Small fuzzy set for sliding surface *s*
Z	:	Zero fuzzy set for sliding surface s
KVL	:	Very Large fuzzy set for control gain *K*
KL	:	Large fuzzy set for control gain *K*
KM	:	Medium fuzzy set for control gain *K*
KS	:	Small fuzzy set for control gain *K*
KVS	:	Very Small fuzzy set for control gain *K*
KZ	:	*Zero fuzzy set for control gain K*

LIST OF SYMBOLS

C	:	dc link capacitor
T	:	time period
T_i	:	switching power devices, i = 1, 2, 3, 4, 5, 6
D_i	:	power diodes, i = 1, 2, 3, 4, 5, 6
V_{dc}	:	dc link Voltage
V_{rms}	:	rms value of inverter output voltage
ω	:	angular frequency
t	:	time variable
V_{un}, V_{vn}, V_{wn}	:	phase voltages
V_{an}, V_{bn}, V_{cn}	:	phase voltages
V_{uv}, V_{vw}, V_{wu}	:	line voltages
V_{ao}, V_{bo}, V_{co}	:	PWM pole voltages
f_c	:	carrier frequency
f_m	:	modulating frequency
f_r	:	frequency ratio
M_r	:	modulation ratio
M_1, M_2	:	variables
T_c	:	carrier period
t_1, t_2	:	time instants
t_k, t_{k+1}	:	time samples
δ_p	:	pulse width
δ_k^p	:	sampled pulse width
$m(t)$	:	modulating signal
δ_k^-, δ_k^+	:	switching instants
ω_s	:	stator frequency in electrical rad/sec
ω_s	:	synchronous frequency in electrical rad/sec

ω_r	:	rotor frequency in electrical rad/sec
ω_{slip}	:	slip frequency in electrical rad/sec
ω_r	:	mechanical rotor speed in rad/sec
ω_{rm}	:	mechanical rotor speed rad/sec
ω_{mr}	:	mechanical rotor speed rad/sec
ω_{sm}	:	synchronous speed in mechanical rad/sec
ω_{ms}	:	synchronous speed in mechanical rad/sec
R_s	:	per-phase resistance of the stator winding
X_s	:	per-phase leakage reactance of the stator winding
L_s	:	per-phase stator self inductance
R_r	:	per-phase resistance of the rotor winding
X_r	:	per-phase leakage reactance of the rotor winding
L_r	:	per-phase rotor self inductance
R_c	:	equivalent resistance of the excitation loss
X_m	:	magnetising reactance
L_m	:	per-phase mutual (magnetising) inductance
I_r	:	rotor current in Amps
I_s	:	stator current in Amps
I_m	:	magnetising current in Amps
S	:	slip speed
P_{in}	:	total input power to the equivalent circuit of the IM in Watt
P_c	:	power representing iron loss in Watt
P_s	:	power representing stator copper loss in Watt
P_r	:	power representing rotor copper loss in Watt
P_g	:	air-gap power in Watt
P_d	:	developed power of the IM in Watt
T_d	:	developed torque in N.m
T_L	:	load torque in N.m
T_d	:	detection period in sec
T_s	:	sampling period in sec
$s\alpha$ - $s\beta$	:	two axis co-ordinates in the stator reference frame
$r\alpha$ - $r\beta$	:	two axis co-ordinates in the rotor reference frame
d - q	:	two axis co-ordinates in the synchronous reference frame
θ_r	:	rotor angle in radians
θ_s	:	stator angle in radians
a	:	spatial operator
f_s	:	general stator variable
f_s	:	clock pulse frequency of the microcontroller
f_{s0}	:	stator zero sequence component
f_{sA}, f_{sB}, f_{sC}	:	three phase stator quantities
$f^s{}_{s\alpha}, f^s{}_{s\beta}$	:	real and imaginary stator components in the stationary reference frame
f_r	:	general rotor variable

f_{ra}, f_{rb}, f_{rc}	:	three phase rotor quantities
$f^{r}_{r\alpha}, f^{r}_{r\beta}$	:	real and imaginary rotor components in the rotor reference frame
$\overline{f_r^s}$	:	rotor variable space vector in the stationary reference frame
$\overline{f_r^r}$	:	rotor variable space vector in the rotor reference frame
$i^s_{s\alpha}$, $i^s_{s\beta}$	:	stator current components in the stationary reference frame
$i^r_{r\alpha}$, $i^r_{r\beta}$	:	rotor current components in the rotor reference frame
$i^s_{r\alpha}$, $i^s_{r\beta}$	:	rotor current components in the stationary reference frame
$\psi^s_{s\alpha}$, $\psi^s_{s\beta}$	:	stator flux linkage component in the stationary reference frame
$\psi^r_{r\alpha}$, $\psi^r_{r\beta}$	:	rotor flux linkage component in the rotor reference frame
$\psi^s_{r\alpha}$, $\psi^s_{r\beta}$	:	rotor flux linkage component in the stationary reference frame
t_l	:	load torque
P	:	number of pole pair
$v^s_{s\alpha}$, $v^s_{s\beta}$	:	stator voltage components in the stationary reference frame
$v^r_{r\alpha}$, $v^r_{r\beta}$	:	rotor voltage components in the rotor reference frame
$v^s_{r\alpha}$, $v^s_{r\beta}$	:	rotor voltage components in the stationary reference frame
ρ	:	derivative operator (d/dt)
J	:	moment of inertia
B	:	damping constant
$\overline{f_s^f}$	:	stator variable space vector in the synchronous reference frame
$\overline{f_r^f}$	:	rotor variable space vector in the synchronous reference frame
v^f_{sd} , v^f_{sq}	:	stator voltage components in the synchronous reference frame
v^f_{rd} , v^f_{rq}	:	rotor voltage components in the synchronous reference frame
i^f_{sd} , i^f_{sq}	:	stator current components in the synchronous reference frame
i^f_{rd} , i^f_{rq}	:	rotor current components in the synchronous reference frame
ψ^f_{sd} , ψ^f_{sq}	:	stator flux linkage component in the synchronous reference frame
ψ^f_{rd} , ψ^f_{rq}	:	rotor flux linkage component in the synchronous reference frame
A, B, C	:	fuzzy sets
U, V, W	:	universe of discourse
μ	:	degree of membership
x, y, z	:	elements in universe of discourse
R	:	fuzzy relation
x_o	:	crisp input value to the fuzzy controller

y_o	:	crisp output value from the fuzzy controller
μ_{agg}	:	aggregate degree of membership
e_o	:	speed error
Δe_o	:	speed error change
k	:	sampling interval
i_{sq}^*	:	torque current command
Δi_{sq}^*	:	torque current command change
s	:	sliding surface
C	:	slop of the sliding surface
u	:	control signal
x_1, x_2	:	system states
φ_1, φ_2	:	controller gain constants
K	:	gain constant
u_R	:	reaching mode control signal
u_S	:	sliding mode control signal
G	:	gain constant represents the control regulator of the induction motor
$\alpha, \beta, \gamma, \xi$	:	controller gains
R	:	reaching mode fuzzy set
S	:	sliding mode fuzzy set
V/f	:	Volts / Hertz scalar control
ϕ_g	:	air-gap flux
K_g	:	winding factor and number of series turns per stator phase constant
E_g	:	air-gap voltage
ω_{ref}	:	reference speed in mechanical rad/sec
ω_{sl}^*	:	slip frequency command in rad/sec
ω_s^*	:	stator frequency command in rad/sec
ω_{fr}	:	rotational frequency in rad/sec
τ_r	:	rotor time constant
K_P, K_I	:	PI controller gains
N	:	constant representing the length of sequence of data
m	:	constant representing the length of successive sub-sequences of data
ω_c	:	cut-off frequency in Hz
ω_{samp}	:	sampling frequency in Hz

Chapter 1, Introduction:

1.1 General introduction:

Various types of electrical machines have been available for over a century, and during this period many research laboratories and industries have made elaborate studies on them. In general, an electrical machine is a complex-structured device that converts its input electrical energy into output mechanical motion. Primitive machines were massive, high-priced, and had deficient dynamic performances. Over the last few decades, however, the evolution of new and enhanced materials together with good comprehension of basic machine principles has virtually contributed to the improvement of machine design (Steven 1983; Sarma 1986; Bose 1988; Murphy and Turnbull 1988; Hubert 1991).

Electrical machines were first utilised in many industrial and domestic applications where only constant speed operation was initially required, such as pumps and fan drives. Later, it becomes necessary to operate these machines in wide range of applications at which variable (adjustable) speed operation is highly demanded. Some of these applications in industry include robots, paper mills, metals-process lines, metals-rolling mills, and general-purpose industrial drives. In the home, applications include washing machines, dryers, air-conditioners, and blenders.

In the past, dc machines have been commonly favoured in areas where a precise variable speed operation is highly required. This is due to the feasible linear control of flux and torque, which is accomplished by simply varying the field and armature currents, respectively. The dc power is normally obtained from a battery or from a rectified ac supply and then controlled by a pulse width modulated chopper to regulate the motor speed, position or torque. Therefore, for high performance applications, the dc machines represent the prime candidate in electrical machines, where a fast and accurate control of both speed and torque can be achieved.

Although the control aspects of dc machines are known to be simple, they are bulky, expensive and have certain disadvantages due to the existence of commutators and brushes. Thus, they do require periodic maintenance and they cannot be used in dusty and explosive environments. Maintenance in general causes inconvenience in unacceptable interruption of operation or when the machine is used in inaccessible locations.

For the preceding reasons, ac machines particularly the squirrel cage induction type have emerged as an alternative in variable speed drive applications. This is because of their simpler, rugged and robust rotor construction, which allows a reliable maintenance-free operation. In contrast to dc machines, the squirrel cage induction motors are brushless and their smaller and simpler construction compared with dc machines results in lower cost motors and high power/weight ratio. That is why the squirrel cage induction motor has been considered as the workhorse in industrial drives (Sarma 1986; Bose 1988; Vas 1990).

However, induction motors in general require more complex control schemes than dc motors, because of their highly non-linear dynamic structure with strong dynamic interactions. The reason for this complexity is due to the inherent dynamic coupling between the direct and quadrature axis of the machine variables (Bose 1988). Furthermore, the dynamics of an induction motor are usually modelled by fifth order state space equations, which represent a non-linear multivariable control system. Here, the control inputs are usually voltage and frequency and the outputs can be speed, position, torque, stator currents or a combination of these.

Over the last few years, this scenario has been changed dramatically with the advent of micro-controllers along with fast switching power converters, which offer more efficient and compact power solutions. Advantages of micro-controllers include significant cost reduction, improved reliability, strong capability of complex computation, and provision of flexible software control (Gabriel et al. 1980; Bowes and Midoun 1988; Addoweesh et al. 1989).

The regulation of utility voltage and frequency, which is responsible for speed regulation, is usually done through an adjustable-frequency inverter circuit. In most industrial applications, the inverter obtains its dc input via uncontrolled voltage rectification. The main function of its basic operation is to change the dc input voltage to sinusoidal balanced 3-phase ac output voltages of desired magnitude and frequency. The output voltage could be variable or fixed at variable or fixed frequency. However, in ac speed drive systems, the output voltage must be varied in conjunction with the frequency at a constant rate to keep the air-gap flux constant (Murphy and Turnbull 1988; Rashid 1993; Mohan et al. 1995).

Variable output voltage can be obtained by varying the gain of the inverter whereas the frequency is determined by the rate at which the semiconductor devices are switched on and off. This can be accomplished by the use of Pulse Width Modulation (PWM) within the inverter control circuitry as shall be explained in Chapter 2.

Various control techniques of varying degrees of complexity have made possible the utilisation of induction motors in high performance speed drive applications where traditionally only dc motors were previously available. The selection of a suitable technique depends on the nature of the control application. The most widely accepted control techniques in ac drive systems are the '*scalar control*' (*V/f*) and the '*vector control*' (also known as field oriented control). Scalar control is a term used to include all general non-vector control schemes, in which the control variables are dc quantities and only their magnitudes are controlled. Such a scheme is characterised by its simplicity and its capability in providing satisfactory steady state behaviour of the induction motor in both open loop and closed loop control (Athani and Deshpande 1980; Bose 1988; Gastli and Matsui 1992; Koga et al. 1992).

However, in the scalar control methods for induction motors, the motor model is considered just for a precise steady state operation to provide a satisfactory steady state performance. Therefore, using this type of control strategy, it is expected that the best dynamic performance of the induction motor can not be achieved during transient operation. This is due to the highly non-linear coupled characteristics of the induction motor. Thus, scalar control methods are bound to be unsuitable for induction motor control when applied in high-performance applications (Bose 1982; Bose 1988; Murphy and Turnbull 1988; Hubert 1991).

An improved control method can be designed to de-couple the control of the two components of the stator current. One component i_{sd} provides the air-gap flux, while the other i_{sq} produces the electromagnetic torque, and as a result an independent control of flux and torque is achieved, which is similar to the control principles of a dc machine. Such control method builds upon the good steady state performance obtained with the aforementioned scalar control, and can give excellent transient dynamic characteristics. The underlying principle of this improved method is known as vector or field oriented control (FOC) (Gabriel et al. 1980; Vas 1990; Liaw and Wang 1991; Liaw et al.; 1991Miki et al 1991; Tez 1995).

In practical applications, however, the de-coupling control between flux and torque can not be maintained for an extensive operation of the speed drive system. This is because of the inevitable parametric perturbation of the induction motor due to flux saturation and heating effects. As a result both steady state and transient responses of the induction motor are degraded. In many speed drive applications which incorporate either scalar or vector control, the main objective of the speed controller are: 1) insensitivity to parameters variation, 2) quick tracking performance of command speed changes without overshoot, and 3) robustness against external load disturbances.

Conventional controllers, for instance the classical PID with fixed gains can work well for the case of linear control, but they cannot generate an optimal response in a plant parameter varying system such as variable speed drive systems. Thus, performance will degrade solely because of the system non-linearity and parameters variation. In other words, the controller gains have to be frequently re-adjusted to cope with any sudden changes occurring in the controlled system.

With the evolution of modern control techniques such as "*fuzzy logic control*" (FLC) and "*variable structure control with sliding mode*" (VSC-SLM), it becomes possible to achieve a robust and improved indirect FOC that is less sensitive to parameter variations. In the past few years, these two types of control schemes have been successfully applied in versatile control application in which robustness against parametric variations and external disturbances are guaranteed (Mamdani 1974; Pappis and Mamdani 1977; Sabanovic and Izosimov 1981; Lin and Tsai 1984; Harashima et al. 1985; Nishimoto et al. 1987; Sabanovic and Bilalovoic 1989; Xu et al. 1989; Chang et al. 1990; Ho and Sen 1990; Ho and Sen 1991; Lim et al. 1991; Sousa et al. 1995; Guillemin 1996; Lai et al. 1996; Da Silva and Acarnley 1997; Vas et al. 1997; Wang and Liaw 1997). Thus, they can be effectively employed in controlled systems with uncertain and time-varying parameters for example in variable speed applications.

In recent years, fuzzy logic has emerged as one of the most attractive control algorithms used in many industrial and domestic applications in the absence of complete and precise mathematical models. Its essence lies in its ease of implementation, as it usually needs no mathematical model of the controlled system for its design. However, a graphical representation of the system behaviour is essential in its initial design, which is also required for the derivation of the fuzzy linguistic rules.

Fuzzy logic is mainly based on a logical model, which simulates the thinking process of a human operator while controlling the system manually. It deals with objects in terms of degree of membership with all possible grades of logic between 0 and 1, and the shades of grey between white and black (Zadeh 1965; Zadeh 1973a).

On the other hand, the theory of variable structure control with sliding mode was first launched in the 1950's. Since then it has been further developed and received worldwide recognition for controlling electrical drives (Sabanovic and Izosimov 1981; Sabanovic and Bilalovic 1989; Xu et al. 1989; Chang et al. 1990; Ho and Sen 1990; Ho and Sen 1991; Lim et al. 1991). It has been reported in the literature that it is capable of providing a fast dynamic response insensitive to system parameters variation and external disturbances. The core role of this control technique is to force the system states of the drive system (error and its derivative) to a pre-determined manifold surface, known as the *sliding surface* (Utkin 1977; Gao and Hung 1993; Hung et al. 1993). The control law is then designed so that the system states always reach the sliding surface and maintain sliding on it towards the origin where the speed error becomes zero.

1.2 Outline of the research work:

The main objective of this research work is to design a relatively simple, cost effective and high performance speed controller for a variable speed induction motor drive. Moreover, the proposed controller should be robust in terms of speed tracking, load disturbance rejection, and parameter variations without the need for complex control algorithms. For this purpose, comprehensive investigation of state-of-the-art modern control systems, which include fuzzy logic and sliding mode control are discussed.

The work is also extended to investigate the possibility of merging the basic algorithms of fuzzy logic into the design of sliding mode control system, which leads to the development of a new integrated control technique. This attempt has been considered in order to improve the dynamic response of the speed drive system and to reduce the inherent chattering effects of sliding mode control (Chang et al. 1990; Ho and Sen 1991; Gao and Hung 1993; Ishigame et al. 1993; Utkin 1993; Chung et al. 1995; Liu and Lin 1996; He et al 1997).

For the development of such speed controllers, it is advantageous to test their effectiveness and validity when applied to the outer-loop speed control of ac drives through computer modeling and simulation. Indeed, this represents a very important step to initially design an efficient controller for the closed loop speed control of an induction motor, in which the design can be tested without costly and time-consuming experiments with hardware. In the present work, computer simulations have been carried out with the simulation package MATLAB and SIMULINK (SIMULINK User's Guide 1992; Brian and Breiner 1995; Leonard and Levine 1995).

For a comprehensive transient study of the drive system, both electrical and mechanical transient dynamics must be considered in the simulation of the induction motor. Therefore, the induction motor model has been created using a set of dynamic equations, which are formulated in the synchronous reference frame as shall be given in Chapter 3 (Krause and Thomas 1965; Lipo et al 1969; Steven 1983; Bose 1988; Vas 1990; Tez 1995). Advantages of this model include highlighting the essential dynamic features of the motor performance in the transient and steady state operation and allowing for much deeper insight into the dynamic behaviour of many motor variables than that in practice.

A brief review of the contents of the thesis will now be given. In chapter 2, elaborate descriptions of voltage source inverter types for single phase and three phase applications are presented. Their operation principles along with the resulted output waveforms and the switching sequences of the power devices are included. An in-depth study of various sinusoidal pulse width modulation techniques for both analogue and microprocessor based implementations is also presented in this chapter. This chapter is referred to frequently throughout the thesis.

In Chapter 3, equivalent circuit and the dynamic models of the induction motor are developed for the steady state and transient response analysis, respectively. The space vectors of various motor quantities (stator and rotor *m.m.f.s*, stator and rotor flux linkages, etc.) are reviewed. This is followed by the derivation of motor dynamic model in both stationary and d-q synchronously rotating reference frames, respectively. The latter serves the purpose of understanding the fundamental principles involved in vector (field-oriented) controlled speed drives. Computer simulations of the motor dynamics are also presented and the obtained result form the basis of further dynamical analysis.

In Chapter 4, a review of recent applications of fuzzy logic control (FLC) in ac speed drive systems is presented. This is followed by a brief summary of some of the relevant concepts in fuzzy set theory and fuzzy logic. The main ideas underlying the FLC are also reviewed. The concept of solving any control problem by a set of linguistic rules is highlighted. The step-by-step procedures to construct a simple FLC controller to regulate the speed of the induction motor are presented. The control behaviour of FLC is simulated in MATLAB environment, in which the fuzzy rules are obtained according to the control experience of the induction motor drive given by Liaw and Wang (1991).

In Chapter 5, a brief review of recent applications of variable structure control with sliding mode (VSC-SLM) in speed drive systems is presented. The basic theory and general mathematical representation of VSC-SLM are also considered in brief. This is followed by design procedures and stability analysis of the conventional VSC-SLM control structure. Finally, the development of the new integrated Fuzzy Sliding Mode Control (FSLMC) scheme that is based on the concept of *reaching law* is thoroughly discussed.

In chapter 6, the general control concepts regarding the scalar and vector control are briefly reviewed. This is followed by a short discussion of field oriented control techniques with the aid of the transient analysis of the induction motor presented in Chapter 3. The dynamic characteristics of the induction motor using various control algorithms with modern control techniques for the speed regulation are assessed via computer modelling and simulation. The control algorithms used are the fuzzy logic control, the variable structure control with sliding mode and the proposed control scheme that introduces the fuzzy logic theory into the deign of the sliding mode algorithms as given in Chapter 4.

In chapter 7, general description of the experimental set-up for the FLC speed drive system is exhibited. This is followed by a detailed presentation of the software development for the closed loop speed control operation. This also includes the generation of the three-phase sinusoidal PWM waveforms and the techniques of digital speed measurements. The design methods for a digital filter based on the rolling average concept to improve the speed measurement are also highlighted. In addition, flowcharts are presented to demonstrate the basic operation of the developed software.

Chapter 8 starts with a brief investigation of the characteristics of some of the motor variables when the induction motor is subjected to different types of sinusoidal PWM signals for a modified switching frequency, f_s. In addition, the performance of the induction motor under open-loop control test is also presented. This followed by illustrating the dynamic performance of the proposed speed control system, which demonstrates the validity of the designed FLC speed controller. Comparison studies between experimental and simulation results under the same operating conditions are also given.

Finally, Chapter 9 presents the conclusion of the thesis and provides some suggestions and recommendations for further developments.

The main contribution of this thesis is the novel approach to design a sliding mode control system using concepts from fuzzy logic algorithms to alleviate the chattering problems and improve the dynamics of the induction motor drive. Further contributions have also been made in several specific areas, particularly in regards to the dynamics comparison for the drive system between the conventional sliding mode controller, the conventional fuzzy logic controller, and the new fuzzy sliding mode controller. These contributions are discussed in the relevant chapters.

Chapter 2. Bridge inverter and PWM techniques:

2.1 Introduction:

The Rectifier-inverter ac drive systems are widely used in different industrial applications. In most cases, the inverter obtains its dc input through uncontrolled voltage rectification. The function of the inverter is to change a dc input voltage to a symmetrical output voltage of desired magnitude and frequency. The output voltage could be variable or fixed at variable or fixed frequency. A variable output voltage can be obtained by varying the gain of the inverter whereas the switching frequency is determined by the rate at which the semiconductor devices are switched on and off (Bowes 1975; Bowes and Mount 1981; Bowes and Clements 1982; Boost and Ziogas 1988; Bowes and Midoun 1988; addoweesh et al. 1989).

The common inverter circuits can be classified as either voltage-source or current source inverters. In the case of voltage source inverter (VSI), the inverter is supplied by a constant or low impedance dc voltage source such as a battery or a rectifier. On the other hand, the current-source inverter (CSI) is supplied with a controlled current from a dc source with high impedance. In other words, the inverter circuit is called VSI if the input voltage remains constant and CSI if the input current is maintained constant (Murphy and Turnbull 1988; Rashid 1993; Mohan et al. 1995). In the present work, only the VSI inverter type is considered throughout the thesis. The VSI can be used in single-phase or three-phase applications depending on the load requirements as depicted in Figures 2-1 and 2-2 respectively.

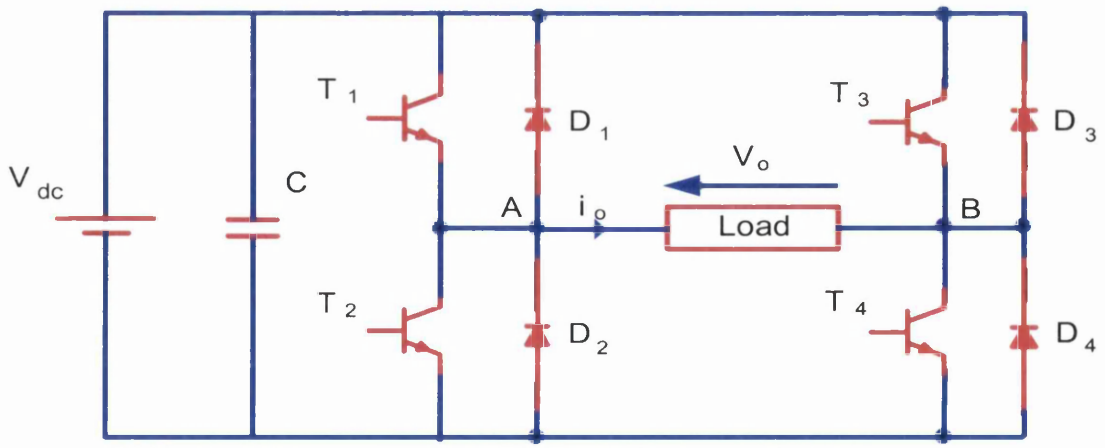


Figure 2-1: Single-phase full-bridge inverter.

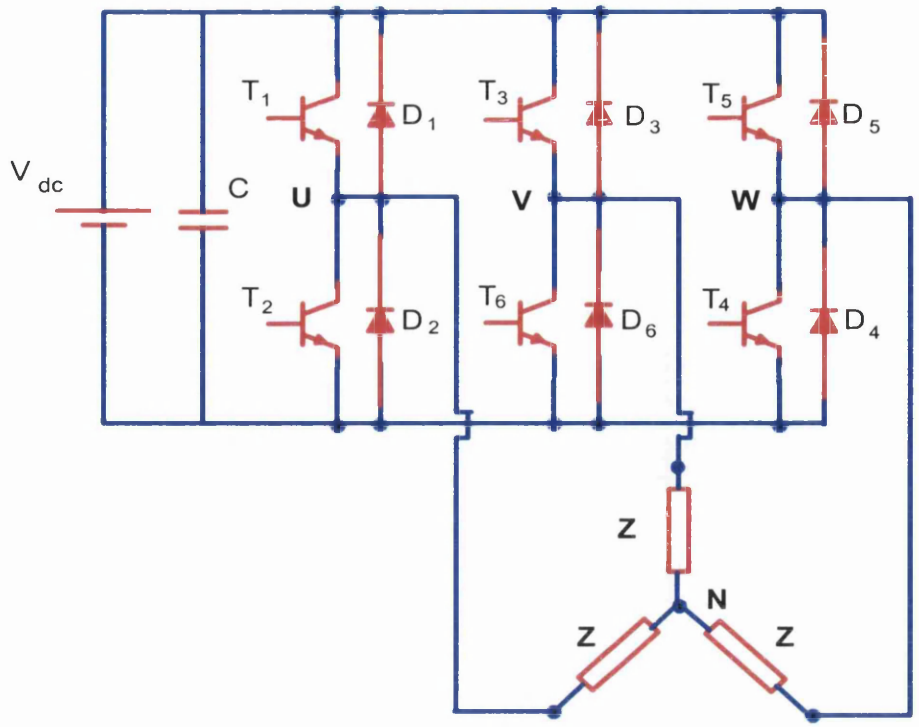


Figure 2-2: Three-phase bridge inverter.

2.1.1 Single phase bridge VSI inverter:

The single-phase inverter introduced in the previous section is shown in Figure 2-1. It consists of four power switching devices acting as choppers. The feedback diodes are connected across the devices to provide a return path for the current in the case of an inductive load. The shunt capacitor C is used to filter out the ac ripple and provides a stiff dc source to the inverter. When transistors T_1 and T_4 are turned on at the same instant of time, the input dc voltage V_{dc} appears across the load. If transistors T_2 and T_3 are switched on, the voltage across the load is reversed and becomes $-V_{dc}$. If the aforementioned sequence of switching is retained the same for alternate 180-degree intervals, an alternate output voltage would appear across the load. The gating signals of the four transistors and the output voltage are shown in Figure 2-3. The rms value of the output voltage is given by

$$V_{rms} = \sqrt{\frac{1}{T}\int_0^T (V_{dc})^2 \, d(\omega t)} = V_{dc} \tag{2.1}$$

For a purely resistive load, the load current has an identical square shape to the output voltage. As a result, the feedback diodes are unnecessary because load current reverses immediately when the output voltage changes polarity.

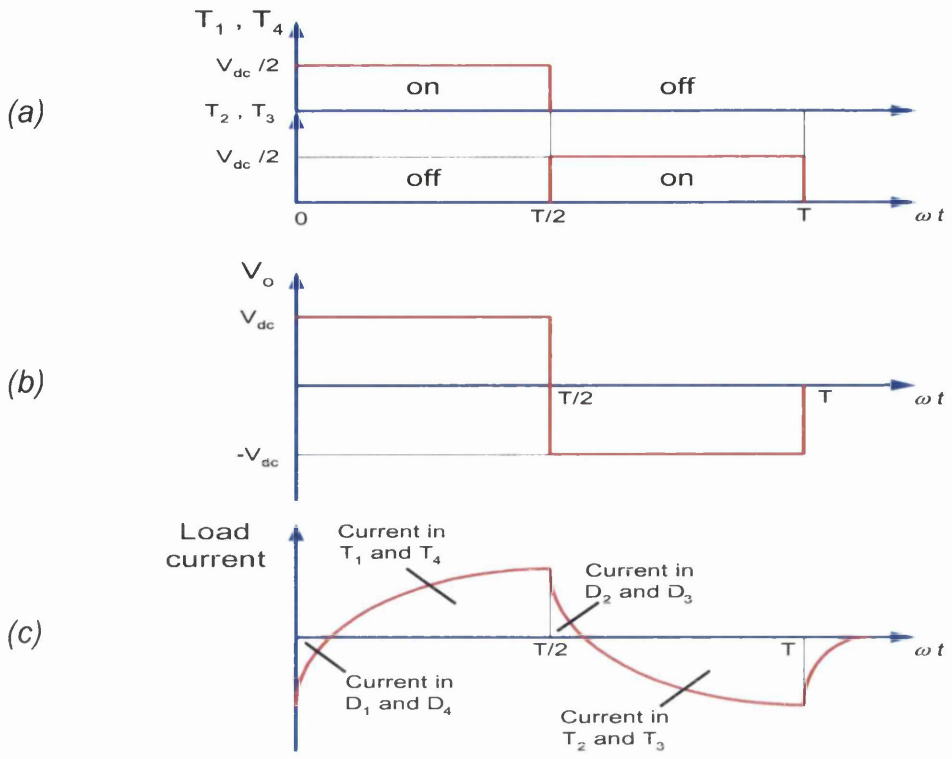


Figure 2-3: Voltage and current waveforms for a single-phase bridge inverter with inductive load; (a) gate signals; (b) load voltage; (c) load current.

However, if the inverter supplies an inductive load, there will be a phase shift between the load current and the load voltage. Figure 2-3 (c) shows the load current waveform in the case of an inductive load. For an interval when the load voltage changes polarity, the load power becomes negative as the voltage and current have opposite signs. This signifies a reverse power flows from the load to the dc supply through the inverter feedback diodes.

2.1.2 Three phase bridge VSI inverter:

The three-phase bridge inverter has been widely used in high power applications such as ac speed drives. Figure 2-2 shows the configuration of the three phase bridge inverter connected to a three phase balanced ac load. A symmetrical three phase output is generated by keeping a phase displacement of 120° between the switching sequences in the three arms of the bridge inverter. Two types of operation can be applied to the transistors: 180° conduction or 120° conduction. Through out the thesis, only the 180° conduction is used.

This mode of operation implies each transistor is to be turned on and off for an interval of 180°. This signifies that each output terminal of the inverter is connected alternately for the same interval to the positive and negative terminals of the dc supply. Three transistors remain on at any instant of time to provide the positive and negative rails for the current. There are six modes of operation in one cycle and the duration of each mode is 60°. Hence the term *three phase six-step VSI inverter.* The gating signals for the transistors in the sequence of T_1, T_2,...,T_6 are shifted from each other by 60° as shown in Figure 2-4.

If this type of inverter is used to feed a balanced star connected load, the phase voltage waveform has six steps per ac cycle and is termed a six-step wave. The resulted phase and line-to-line voltage waveforms in the case of three phase balanced resistive load are depicted in Figure 2-5. However, these waveforms are not load dependent and they are valid for any balanced three-phase linear load or ac motors.

2.2 PWM switching techniques:

The aim of using the inverter circuit is to generate voltage or current waveforms close to sinusoidal shape with controllable magnitude and frequency. However, the basic switching of a practical inverter as in the six-step operation normally results in non-sinusoidal ac waveforms that may severely affects the motor performance. In other words, the generated waveform would suffer the presence of low order harmonics in its spectra. This led to the development of PWM techniques, where not only the fundamental component is controlled but also harmonic content is significantly improved (Bowes 1975; Bowes and Mount 1981; Bowes and Clements 1982; Boost and Ziogas 1988; Bowes and Midoun 1988; addoweesh et al. 1989).

In general, the PWM controls amplitude and frequency of the output ac waveforms and minimizes the harmonic content in the load current. Several PWM techniques have been developed in the past. The most common techniques being applied in ac drives are sinusoidal PWM (SPWM), space vector PWM (SVPWM) and current-impressed or hysteresis PWM (HPWM). The SPWM is used in drive applications where open loop and closed loop control can be performed. Moreover, it is the preferred approach due to the low harmonics waveform characteristics and implementation simplicity (Bowes 1975; Bowes and Mount 1981; Bowes and Clements 1982; Boost and Ziogas 1988).

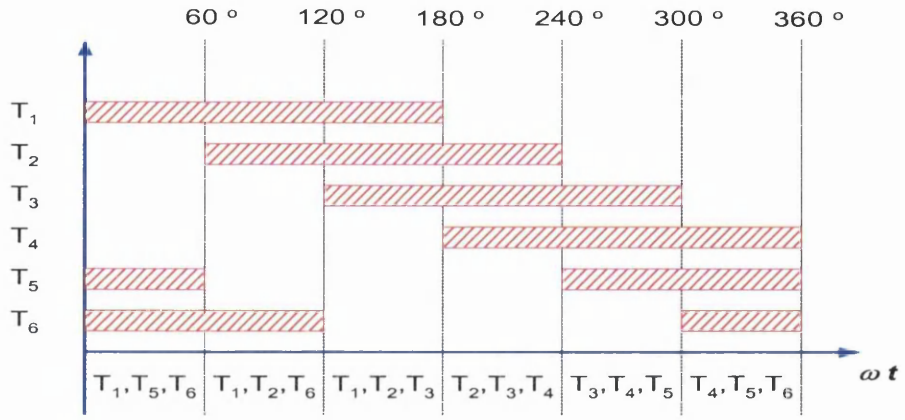
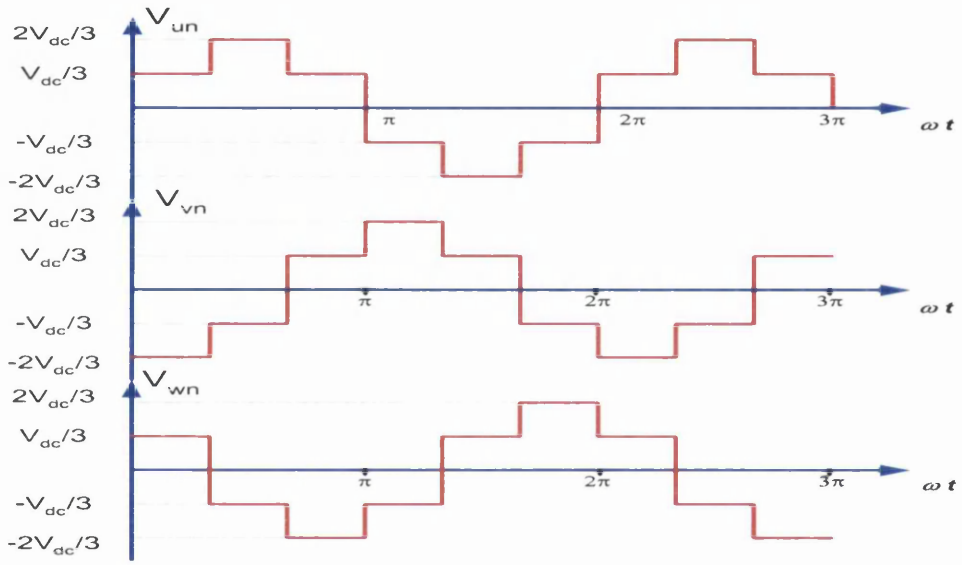


Figure 2-4: Gating signals for 180° operation.

(a)



(b)

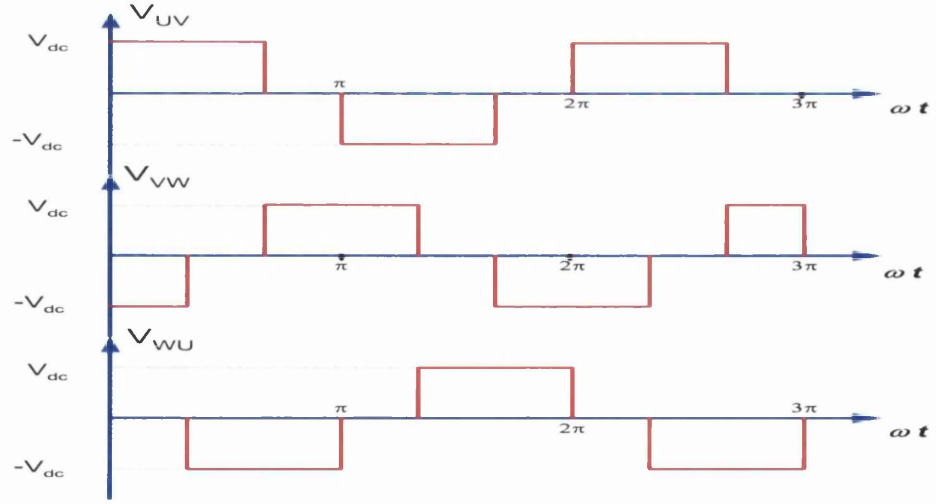


Figure 2-5: Voltage waveforms for six-step VSI inverter with 180° operation; (a) phase voltages; (b) line-to-line voltages.

On the other hand, the SVPWM and HPWM can only be used in closed-loop control because they are dependent on the instantaneous motor behavior. Only the SPWM technique is considered throughout the thesis. The PWM signals are pulse trains with fixed magnitude and variable frequency and pulse widths. The switching instants of the pulses are defined by comparing a reference sinusoidal modulating wave with a triangular carrier wave.

The fundamental output voltage is controlled by varying the duty cycle of the PWM waveform, while changing the frequency of the modulating waveform alters the output frequency. The frequency of the carrier wave should be greater than that of the modulating wave, so that the resulting PWM waveform approaches closely the ideal sinusoidal waveform. Several SPWM techniques have been proposed in the past to improve the source utilization and reduce the harmonic contents in the generated waveforms. In what follows, most common SPWM techniques are described.

2.2.1 Natural sampling SPWM:

In the past, natural sampling SPWM technique has been widely used in ac drive control applications because of its inherent simplicity and ease of implementation using analogue control circuitry. In this type of sampling, as shown in Figure 2-6, the modulating sine wave is compared directly with a high frequency carrier triangular wave and the intersection points define the switching instants of the pulses. The ratio of the carrier frequency f_c to the modulating frequency f_m is termed frequency ratio f_r, which determines the number of pulses per ac cycle. The amplitude of the carrier wave is fixed, while the amplitude of the modulating wave is variable.

The ratio of the modulating amplitude to the carrier amplitude is known as the modulation ratio M_r. The rms output voltage is controlled by changing the pulse widths, which are altered as a result of varying the modulation ratio M_r. As for the harmonic contents in the output voltage waveform, the PWM pushes the harmonics into higher frequency range around the carrier frequency f_c. In other words, the predominant harmonics would occur as side bands of the carrier frequency f_c and its multiples (Murphy and Turnbull 1988; Rashid 1993; Mohan et al. 1995).

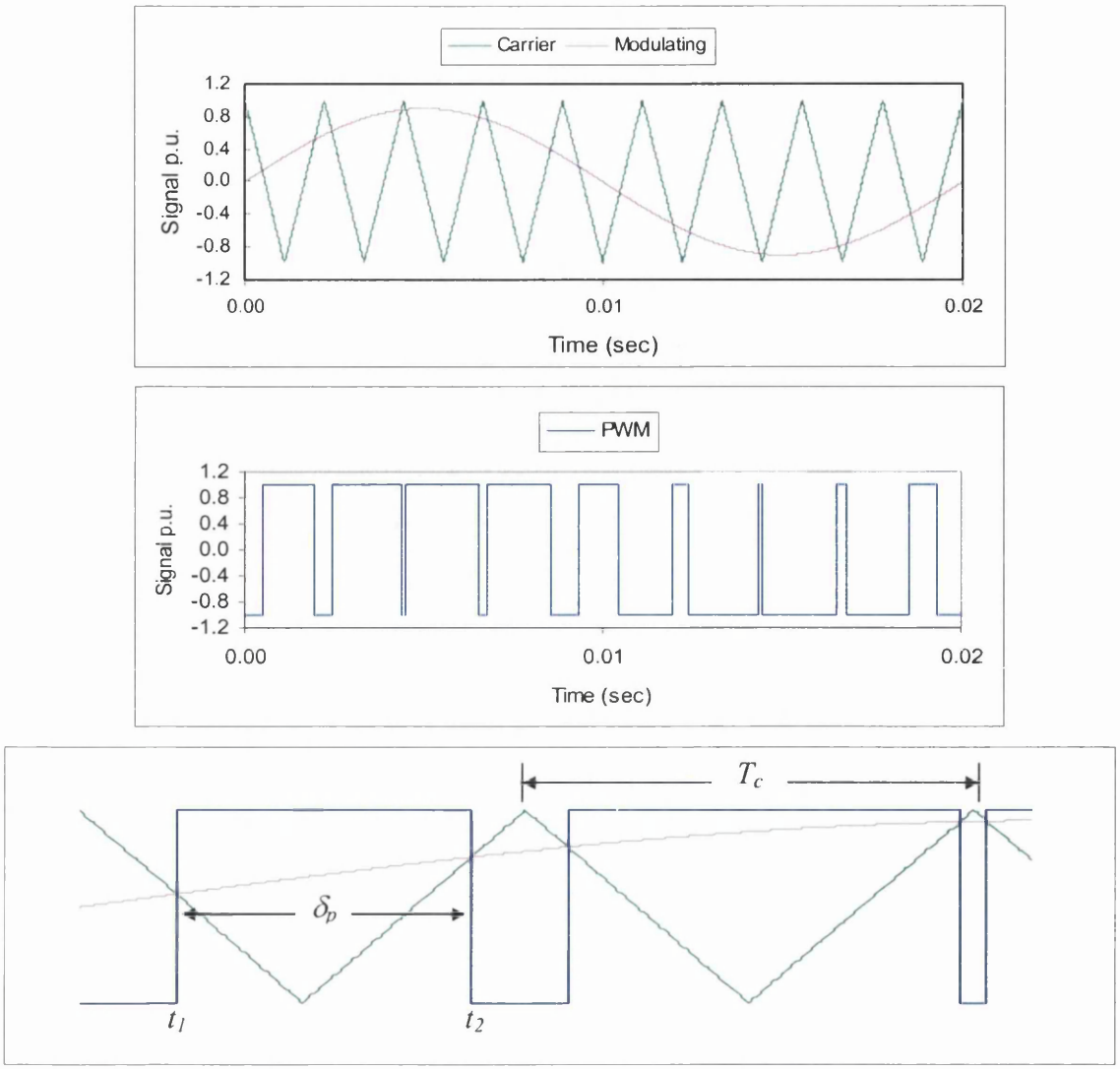


Figure 2-6: 2-level natural sampling generation

As stated before, this type of sampling is only appropriate for analogue implementation and therefore it is not possible to define the pulse widths using direct analytical expression. That is because the switching instants are defined by the instantaneous intersection of the carrier wave and the modulating wave. As illustrated in Figure 2-6, the modulating wave is varying while the sampling process is taking place. This means that the pulse width is proportional to the height of the modulating wave at the instant when switching occurs (t_1 and t_2) (Bowes 1975; Bowes and Clements 1982). Hence, the centers of the resulted pulses are not uniformly spaced. However, according to Bowes and Clements (1982) the pulse width can be defined using a transcendental equation of the form,

$$\delta_p = \frac{T_C}{2}\left[1+\frac{M_r}{2}\{\sin(\omega t_1)+\sin(\omega t_2)\}\right] \tag{2.2}$$

where T_c is the carrier period. Also shown by Bowes and Clements (1982) is that the switching instants of the pulses are defined via a set of nonlinear equations. Consequently, these equations can be simply solved using numerical techniques such as Newton-Raphson. It should be noted that the PWM waveforms shown in Figure 2-6 swing between two voltage levels +1 and –1, and therefore it is usually referred to as *2-level* PWM waveform (Bowes 1975; Bowes and Clements 1982; Mohan et al. 1995). On the other hand, it is possible to produce a *3-level* PWM waveform by switching between +1, 0 and –1 as shown in Figure 2-7.

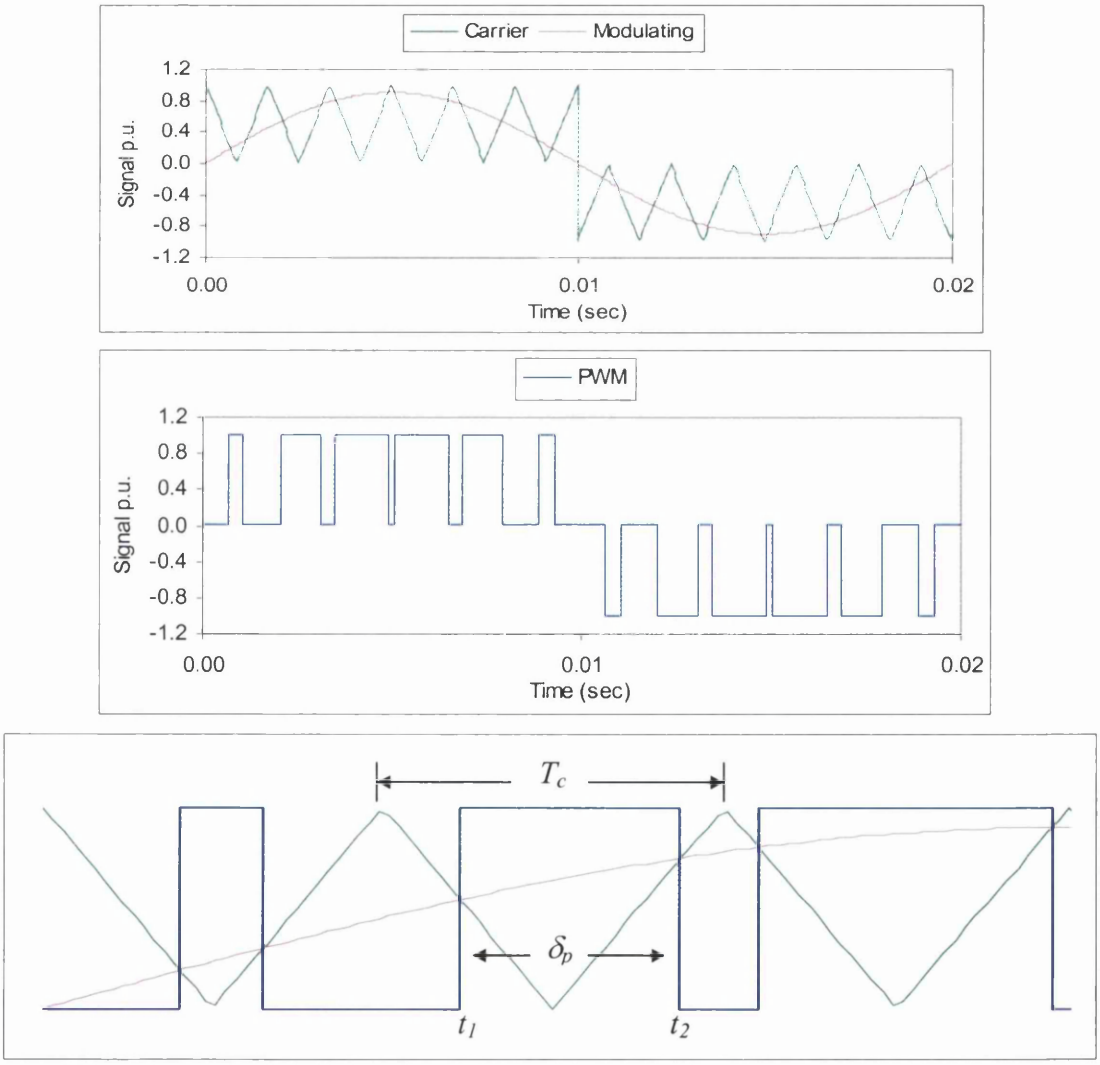


Figure 2-7: 3-level natural sampling generation

The pulse width can be defined using a transcendental equation of the form,

$$\delta_p = \frac{T_C}{2} M_r \{\sin(\omega t_1) + \sin(\omega t_2)\} \tag{2.3}$$

A wide variety of SPWM techniques can be derived from the foregoing 2-level and 3-level natural sampling generation in order to improve the overall system performances and the generated input and output harmonics content. Some of these techniques will be discussed in the following sections.

2.2.2 Third-harmonic injection natural sampling SPWM:

This is an improved technique derived from the natural sampling SPWM through the injection of approximately 17% third harmonic component to the original modulating wave. This results in a nearly flat-topped modulating wave that increases the fundamental output voltage by approximately 20%, while maintaining a low harmonic distortion (Boost and Ziogas 1988; Bowes and Midoun 1988). The modulating wave with third harmonic injection is shown in Figure 2-8. According to Boost and Ziogas (1988) the hardware implementation of this technique is quite simple. It should be noted that the addition of the third harmonic to the original modulating signal does not affect the quality of the output voltage because the output of the three phase inverter does not contain triplen harmonics. The modulating wave would be of the form,

$$m(t) = M_r\,[\sin(\omega t) + 0.17\sin(3\omega t)] \tag{2.4}$$

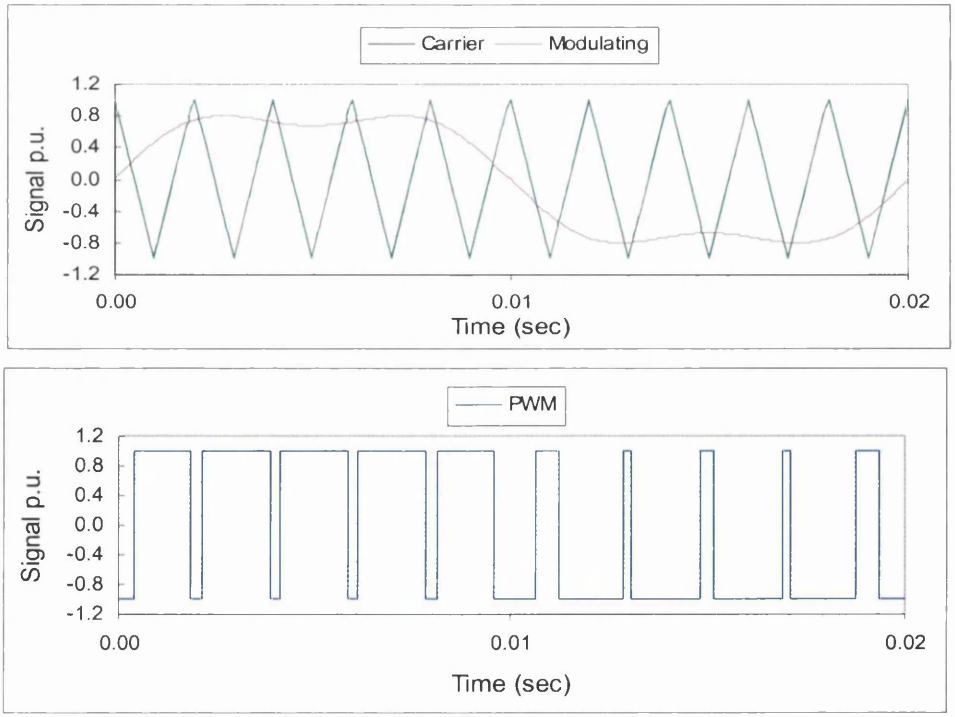


Figure 2-8: 2-level 3rd harmonic injection natural sampling SPWM generation.

2.2.3 Modified SPWM:

In theory, increasing the number of pulses per ac cycle reduces the harmonic content and this can be achieved by increasing the carrier frequency. However, due to the increased number of switching ON and OFF states, it would cause tremendous switching stresses on the power devices, which results in higher switching losses. This in turn will shorten the life durability of the power devices.

An improved PWM switching strategy has emerged, where the basic SPWM technique can be modified so that the carrier wave is only applied in the first and last 60° intervals per half cycle of the output ac waveform, while the 60° to 120° intervals are kept unmodulated on. The main features are depicted in Figure 2-9. In this type of modulation, the fundamental component is higher compared to that of the SPWM technique and inherits the possibility of producing less switching pulses, while using a high carrier frequency. Therefore, it provides less switching stresses on the devices and hence the switching losses are reduced. However, it generates low order harmonic component in the output waveform due to the unmodulated interval between 60° and 120° (also between 240° and 300°) (Boost and Ziogas 1988; Rashid 1993).

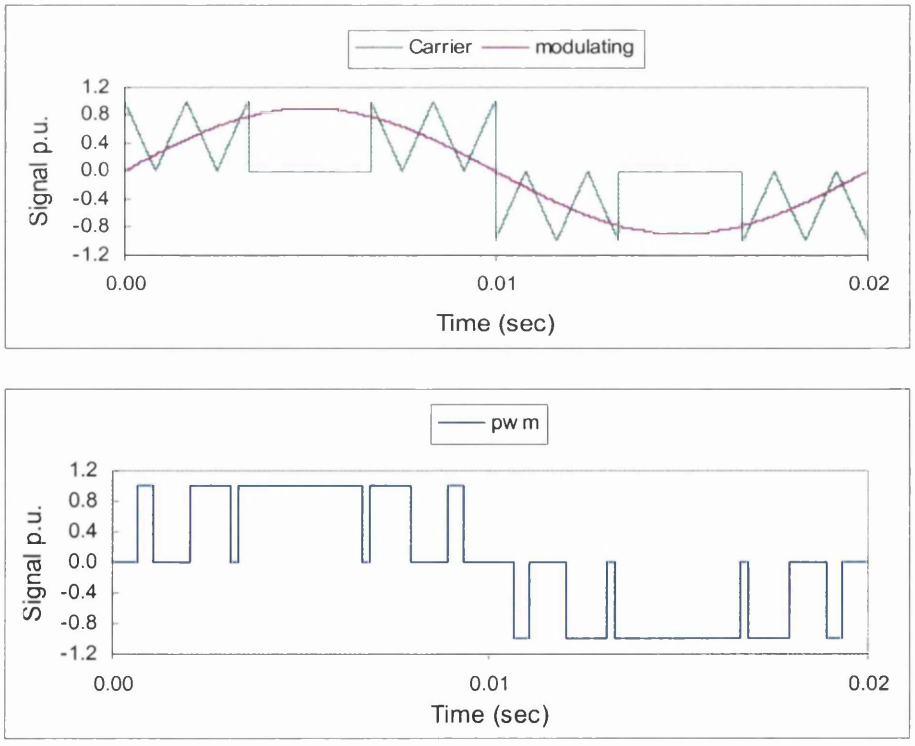
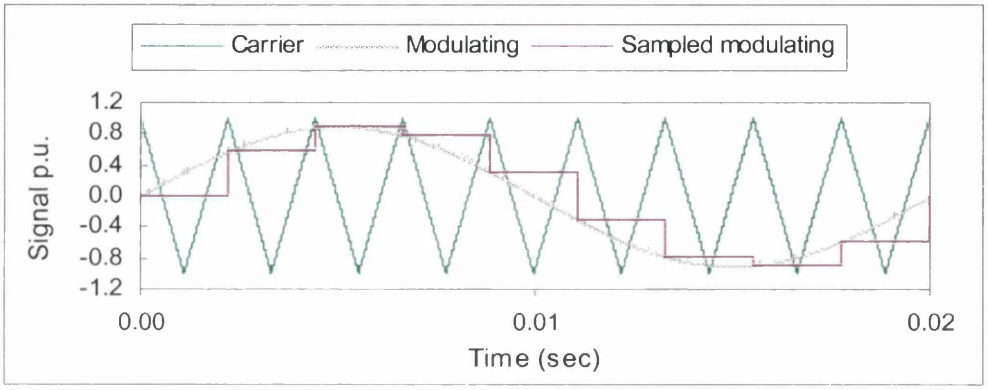
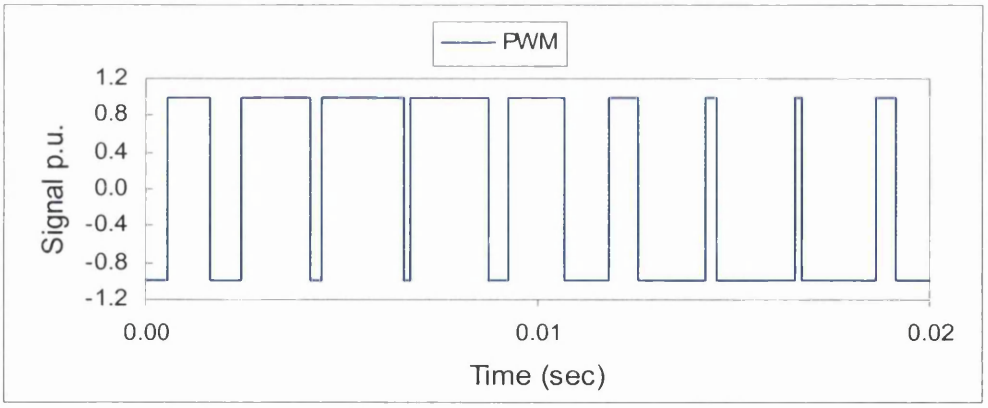


Figure 2-9: 3- level modified natural sampling SPWM generation.

2.2.4 Regular sampled SPWM:

In recent years, there have been growing trends towards the use of digital circuitry and microprocessor-based controllers for the generation of PWM techniques. This results in considerable reduction in hardware implications and offers flexibility in generating different types of PWM techniques by changing only the software. However, all the natural sampling SPWM techniques described so far are essentially analogue in nature, which means that their implementation in digital form is tedious, unless a very fast microprocessor is used.





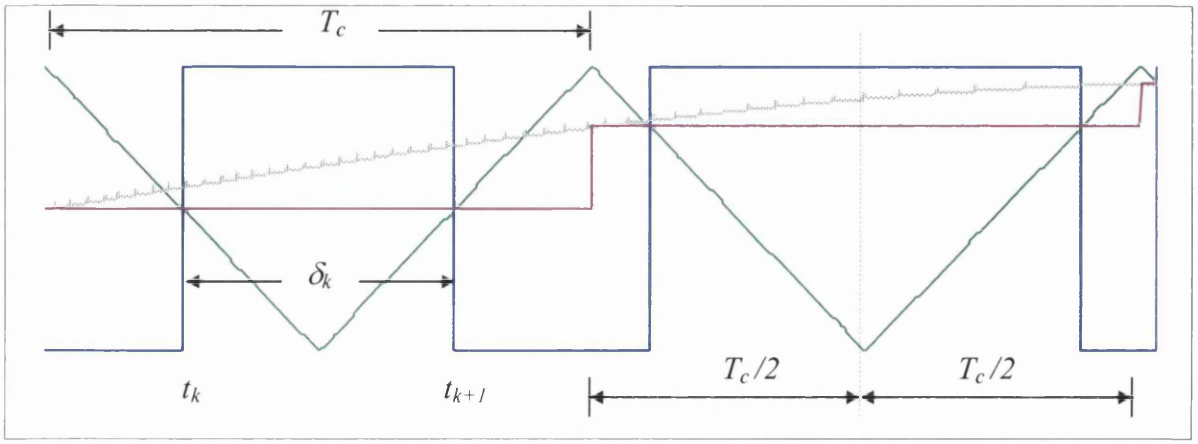


Figure 2-10: 2-level regular symmetric SPWM generation.

The regular sampled SPWM technique has emerged as an alternative approach that is suitable for digital and microprocessor implementation. In this type of sampling, the modulating sine wave is sampled at regular intervals every carrier period T_c as illustrated in Figure 2-10. This sampling process is known as *Regular Symmetric SPWM*. The level of each sample is maintained constant until the next sample occurs. Therefore, a stepped modulating signal is constructed as a sampled approximation of the original sine signal. This sampled signal is then compared with the triangular carrier signal and the intersection points define the switching instants of the SPWM pulses. In contrast to the natural sampling technique, the pulse width of the regular sampled SPWM can be analytically defined by solving the switching instants occurred at the falling and rising edges of the triangular carrier wave as follows: the switching instant for the falling edge is given by,

$$\delta_k^- = \frac{T_C}{4}\left[1 - M_r \sin(\omega t_k)\right] \tag{2.5}$$

and similarly solving for the rising edge yields,

$$\delta_k^+ = \frac{T_C}{4}\left[3 + M_r \sin(\omega t_k)\right] \tag{2.6}$$

Therefore, the pulse width is defined by,

$$\begin{aligned} \delta_k^p &= \delta_k^+ - \delta_k^- \\ &= \frac{T_C}{2}\left[1 + M_r \sin(\omega t_k)\right] \end{aligned} \tag{2.7}$$

where T_c is the carrier period which equals the sampling interval, and t_k is the sampling instant. In addition to this type of sampled modulation, the modulating sine wave can be sampled twice at equal intervals every carrier period T_c. The main intrinsic features of this type are shown in Figure 2-11. This sampling process is called *Regular Asymmetric SPWM*. Here, the pulse width is determined using two samples of the modulating wave, while only one sample is used to determine the pulse width in the symmetric SPWM. According to Abed (1995) the asymmetric SWPM generates fewer harmonics in the output voltage and current waveforms compared to the symmetric SPWM. However, for microprocessor software implementation, the computation overhead required to generate the asymmetric SPWM is doubled compared to that of the symmetric SPWM.

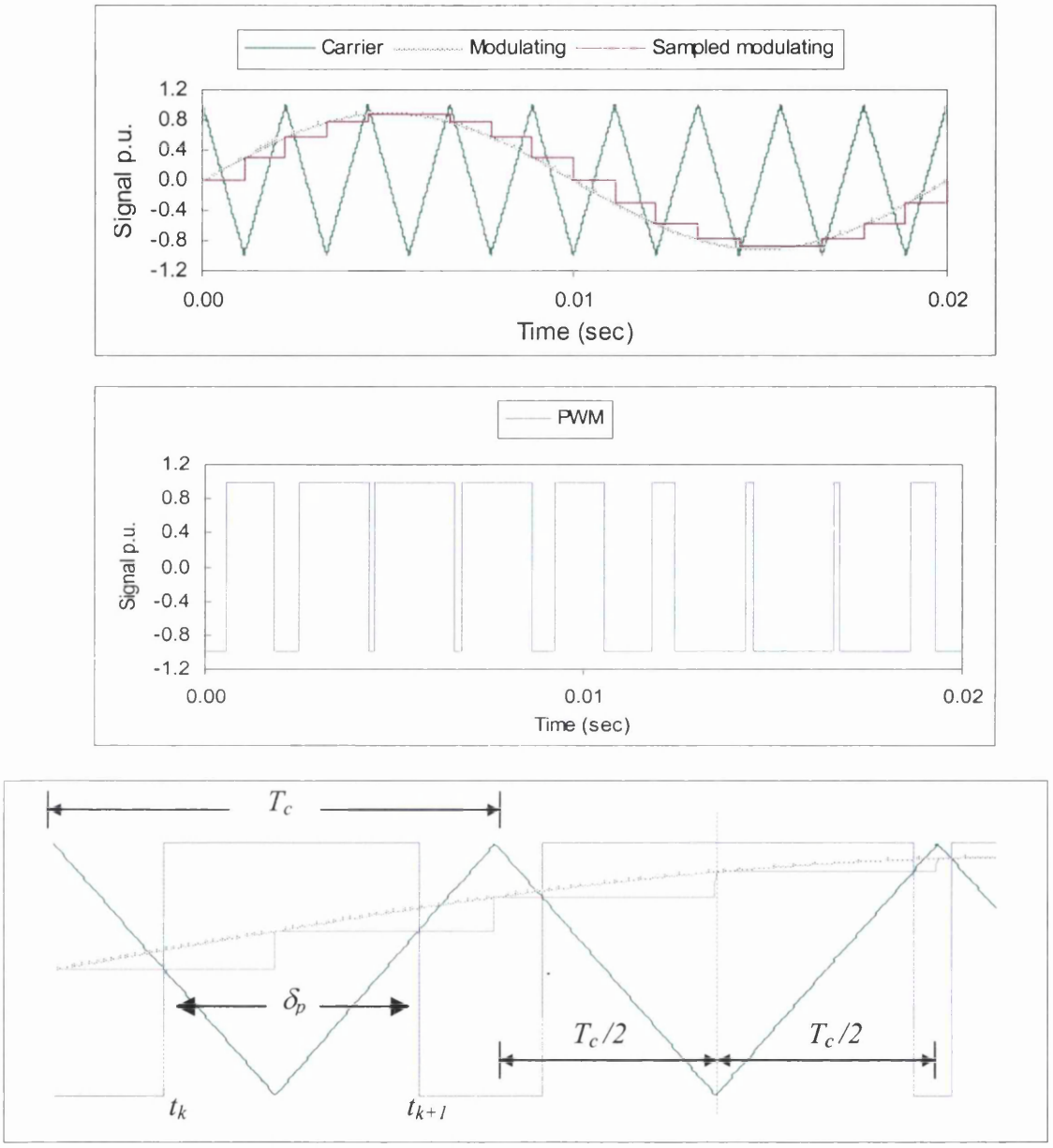


Figure 2-11: 2-level regular asymmetric SPWM generation.

As stated before, the pulse width in the asymmetric SPWM is determined using two different samples of the modulating wave taken at time instants t_k and t_{k+1}. These samples define the leading and falling edges of each pulse respectively. Consequently, the pulse width can be defined as follows: for the falling edge of the triangular carrier wave, the switching instant is given by,

$$\delta_k^- = \frac{T_C}{4}\left[1 - M_r \sin(\omega t_k)\right] \tag{2.8}$$

and similarly the switching instant for the rising edge is,

$$\delta_k^+ = \frac{T_C}{4}\left[3 + M_r \sin(\omega t_{k+1})\right] \tag{2.9}$$

Therefore, the pulse width is defined by,

$$\begin{aligned} \delta_k^p &= \delta_k^+ - \delta_k^- \\ &= \frac{T_C}{2}\left[1 + 0.5 * M_r \{\sin(\omega t_k) + \sin(\omega t_{k+1})\}\right] \end{aligned} \tag{2.10}$$

where,

$$t_{k+1} = t_k + \frac{T_C}{2} \tag{2.11}$$

2.3 Summary:

In this chapter, the types of voltage source inverters for single phase and three phase applications were introduced. The reminder of the chapter then concentrated on various sinusoidal pulse width modulation (SPWM) techniques for both analogue and microprocessor based implementations. The following are the main points made in this chapter:

1. An inverter circuit can be classified as either voltage-source or current source inverter. In the case of voltage source inverter (VSI), the inverter is supplied by a constant or low impedance dc voltage source such as a battery or a rectifier. On the other hand, the current-source inverter (CSI) is supplied with a controlled current from a dc source with high impedance.

2. A six-step inverter type normally results in non-sinusoidal ac waveform that suffers from the presence of low order harmonics in its spectra, which may severely affects the motor performance. This led to the development of PWM techniques, where not only the fundamental component is controlled but also harmonic content is significantly improved.

3. There are several SPWM techniques to improve the source utilization and reduce the harmonic contents in the generated current and voltage waveforms. These techniques can be derived from 2-level or 3-level natural sampling modulation.

4. The natural sampling modulation is essentially analogue in nature, which means that its implementation in digital form is tedious, unless a very fast microprocessor is used. An alternative approach has emerged that is suitable for digital and microprocessor implementation, known as symmetric or asymmetric regular sampled SPWM technique. The 2-level asymmetric technique is used throughout the thesis.

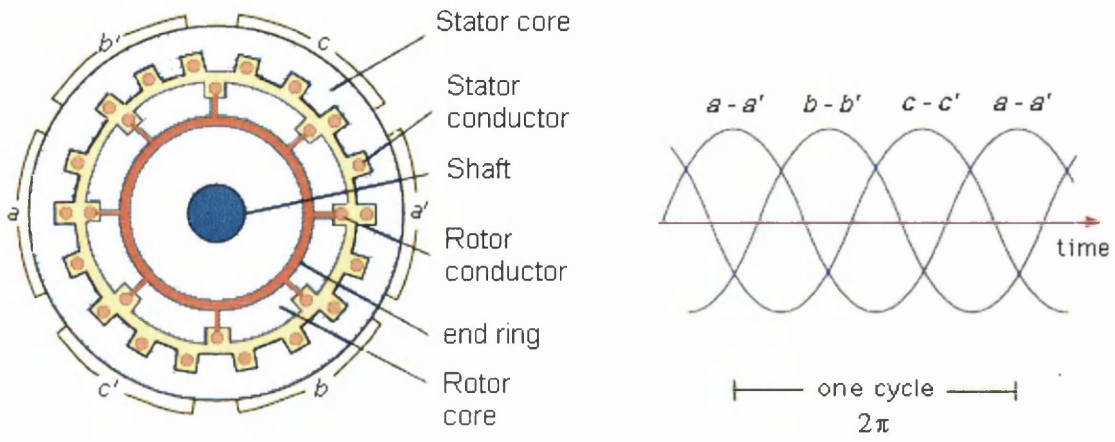
Chapter 3, Modeling and simulation of ac induction motor:

3.1 Introduction:

The three phase induction motors have been for many years regarded as the workhorse of industry due to their simple, rigid and economical construction. Further, they are lightweight, robust and of low maintenance compared to dc motors. They have been extensively used in many drive applications where variable speed operation is essential. The least expensive and the most widely spread induction motor is the squirrel cage type due to simplicity and reliability of the cage construction (Sarma 1986; Bose 1988; Vas 1990; Hubert 1991).

The wires along the rotor axis are connected by a metal ring, termed the *end-ring*, resulting in a short circuit. When the current flows in the stator windings, it creates a magnetic field in the air-gap rotating at *stator frequency* ω_s (also known as *synchronous frequency*). This field induces current in the cage windings, which then produces another magnetic field around the rotor circuit. The interaction between the stator and the rotor magnetic fields results in a steady torque, which allows the rotor to start revolving in the direction of the stator field but at a slower speed that is related to the *rotor frequency* ω_r. The difference between the stator and the rotor frequency is termed *slip frequency* ($\omega_{slip} = \omega_s - \omega_r$) (Jones 1967; Sarma 1986; Hubert 1991; Mohan et al.1995).

In this chapter, equivalent circuit and dynamic model of the induction motor are developed for the steady state and transient response analysis, respectively. The space vectors of various motor quantities (stator and rotor *m.m.f.s*, stator and rotor flux linkages, etc.) are reviewed. This is followed by the derivation of motor dynamic model in both stationary and d-q synchronously rotating reference frames. Computer simulations of the motor dynamic model carried out in MATLAB and SIMULINK are presented and the obtained results formed the basis of further dynamic analysis. In the subsequent sections, the subscript *s* and *r* represent the stator and rotor variables, respectively. On the other hand, the superscript *s*, *r*, and *f* represent the stator, rotor and synchronous reference frames in dynamic modeling, respectively.



(a) *(b)*

Figure 3-1: Three phase two-pole squirrel cage induction motor: (a) cutaway view of the motor, (b) the resultant sinusoidal three phase stator currents.

3.2 Steady state equivalent circuit of an AC induction motor:

A cutaway view of a practical three-phase two-pole induction motor is shown in Figure 3-1. The cylindrical stator and rotor cores are furnished with balanced three phase windings. The three-phase winding has three stator coils displaced 120 degrees a part in space. Each phase winding produces a magnetic field or magnetomotive force *mmf* in the air-gap when connected to a balanced three-phase voltage supply. This magnetic field is sinusoidally distributed in space at any instant of time and rotates at synchronous frequency ω_s. Due to the induction effect currents will start to flow in the rotor winding as a result of the stator magnetic field. Consequently, these rotor currents will set up a magnetic field similar to that produced by the stator but rotating at slip frequency ω_{slip}. The electric torque is then developed as a result of the stator and rotor field interactions in the air gap.

The induction motor may then be viewed as a transformer with an air-gap in which the electric power is transferred between the stator and the rotor. The equivalent circuit of an induction motor is quite similar to that of a transformer, where the stator corresponds to the transformer primary and the rotor

corresponds to the secondary. For a balanced steady state analysis, it is sufficient to express the motor parameters on a per-phase basis. Hence, a per-phase equivalent circuit of the induction motor is now developed as depicted in Figure 3-2.

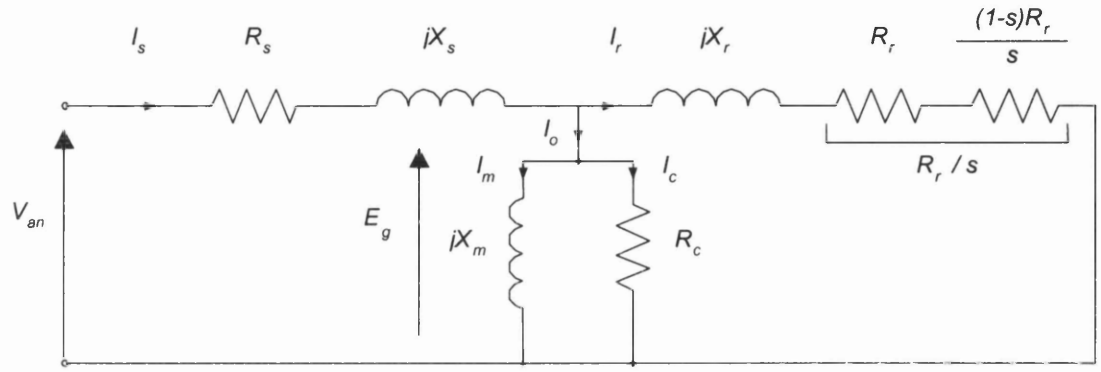


Figure 3-2: Per-phase equivalent circuit of induction motor.

In the above equivalent circuit, R_s and X_s are the per phase resistance and leakage reactance of the stator winding and consequently, $R_s + jX_s$ is termed the stator leakage impedance. Similarly, for the rotor winding, R_r and X_r represent the equivalent rotor resistance and the rotor leakage reactance per-phase, respectively. The excitation (or iron) loss is represented by an equivalent resistance R_c, which is placed in parallel to the magnetizing reactance X_m (Sarma 1986; Hubert 1991).

The mechanical rotor output, which is the cause of the motor running can be considered as an equivalent rotor resistance known as the dynamic resistance, which is of the form R_r [(*1-S)/S*)]. The combined effect of this resistance and the equivalent rotor resistance R_r appears as a reflected resistance R_r / S, where S is the slip speed and can be expressed as a fraction by

$$S = \frac{(\omega_{sm} - \omega_{rm})}{\omega_{sm}} \tag{3.1}$$

where, ω_{sm} and ω_{rm} are referring to the mechanical synchronous and rotor speeds in rad/s, respectively. The total input power to the equivalent circuit is given by

$$P_{in} = P_c + P_s + P_g \tag{3.2}$$

where, P_c is the *iron loss* (= $3I_c^2 R_c$), P_s is the *stator copper loss* (= $3I_s^2 R_s$) and P_g is the *gap power* (=$3I_r^2 R_r /s$). The term gap power corresponds to the power transferred electromagnetically across the air gap between the stator and the rotor. The total electrical loss in the induction motor is defined as

$$P_{loss} = P_c + P_s + P_r \tag{3.3}$$

where, P_r represents the rotor copper loss (= $3I_r^2 R_r$). The difference between the input power and the total losses represents the developed power

$$\begin{aligned} P_d &= P_{in} - P_{loss} \\ &= 3I_r^2 R_r \left(\frac{1-S}{S} \right) \end{aligned} \tag{3.4}$$

The electromagentic torque developed by the motor can be then determined as

$$\begin{aligned} T_d &= \frac{P_d}{\omega_{rm}} \\ &= \frac{3}{\omega_{sm}} I_r^2 \frac{R_r}{S} \end{aligned} \tag{3.5}$$

This equation shows that the developed torque is a function of two variables, which are rotor current I_r and slip S. Hence, two methods of controlling the torque of the induction motor are feasible: current control and slip control.

In general, the per-phase equivalent circuit described here is used to determine various motor quantities (stator and rotor currents, input and output power etc.) under steady state operation of the induction motor. In theory, this model is sufficient to highlight the essential features of the motor performance. However, there exists an alternative model, which is capable of representing the behavior of the induction motor in transient operation, termed *dynamic model.* A detailed development of this model will be presented in the following section.

3.3 Dynamic model of an AC induction motor:

It is preferred to represent the induction motor by a dynamic model to describe its real performance under both transient and steady state operation. Furthermore, it serves the purpose of understanding the fundamental principles involved in *vector controlled* (*field-oriented*) drives. The rotational symmetry of an induction motor may be taken advantage of by vectorial representation in three-phase and two-phase coordinates as shown in Figure 3-3 (Krause and Thomas 1965; Lipo et al. 1969; Lipo and Krause 1969; Steven 1983; Bose 1988; Vas 1990).

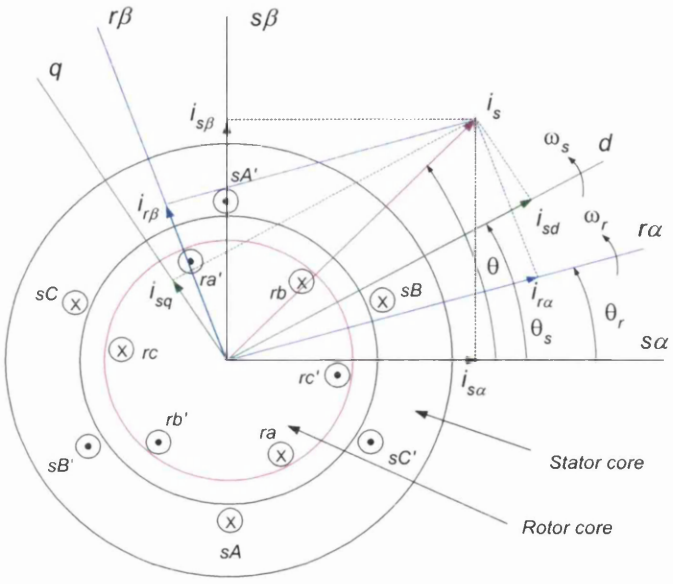


Figure 3.3: Cross-section of symmetrical three-phase machine; relationship between stationary, rotating and synchronous reference frames.

For simplicity, some basic assumptions are made to avoid any mathematical and physical complications in deriving the equations of the dynamic model. These assumptions are:

- Magnetic saturation of induction motor is neglected.
- The machine structure has a smooth (uniform) air-gap with symmetrical two-pole, three phase winding.
- The permeability of the iron parts in the machine is infinite.

Figure 3-3 shows the cross section view of a three-phase two-pole induction motor. The stator and rotor windings are shown as a single, multiple-turn full pitch coils placed on both sides of the air-gap. The figure also shows how three phase induction motor can be analyzed in terms of a complex planes perpendicular to the motor axis. The axis denoted by $s\alpha$-$s\beta$ and $r\alpha$-$r\beta$ represent the static and moving two axis coordinates in the stationary and rotating reference frames, respectively, while d-q represents the synchronously rotating coordinates that is rotating at synchronous frequency ω_s.

Some modeling techniques have been presented in literature, which are based on the selection of desired reference frames (Jordan 1965; Robertson and Hebbar 1969; Lipo and Cosoli 1984; Ho and Sen 1986; Ghani 1988). The most common frames used in dynamic modeling are *stationary reference frame*, *rotor reference frame* and *synchronous reference frame*. In other words, the space vector equations of various motor quantities (stator and rotor voltages, stator and rotor currents, flux linkages etc.) can be formulated referring to different domains.

In the stationary reference frame, the measured three-phase stator quantities are transformed into two-axis ($s\alpha$-$s\beta$) quantities, where they are fixed to the stator and oriented 90° to each other (Bose 1988; Vas 1990; Tez 1995) as illustrated in Figure 3-3. Similarly, the rotor reference frame is where the two-axis quantities denoted by $r\alpha$ and $r\beta$ are fixed to the rotor. The speed of this reference frame is

$$\omega_{mr} = {}^{d\theta_r}\!/_{dt} \tag{3.6}$$

where, ω_{mr} is the mechanical rotor speed (rad/sec), and θ_r is the rotor angle (also shown in Figure 3-3). On the other hand, it is possible to express the space vector equations in a general reference frame where the two-axis (d-q) quantities are rotating at synchronous speed ω_s, hence the term synchronous reference frame. The great advantage of using this frame is that the induction machine would be seen as a separately excited DC machine, resulting in d.c. quantities in steady state.

Currently, the model equations of the induction motor expressed in the synchronous reference frame is the most commonly used model for transient and steady state analysis. Moreover, it forms the basis for designing various forms of control of ac drives such as vector control (Plunkett 1977; Gabriel et al.

1980; Nabae et al. 1980; Bose 1982; Kubo et al. 1985; Leonhard 1986; Mutoh et al. 1986; Leonhard 1988; Wu and Strangas 1988; Takahashi and Ohmori 1989). In what follows, derivations of induction motor model in the aforementioned reference frames are presented and the relationship between them is also reviewed.

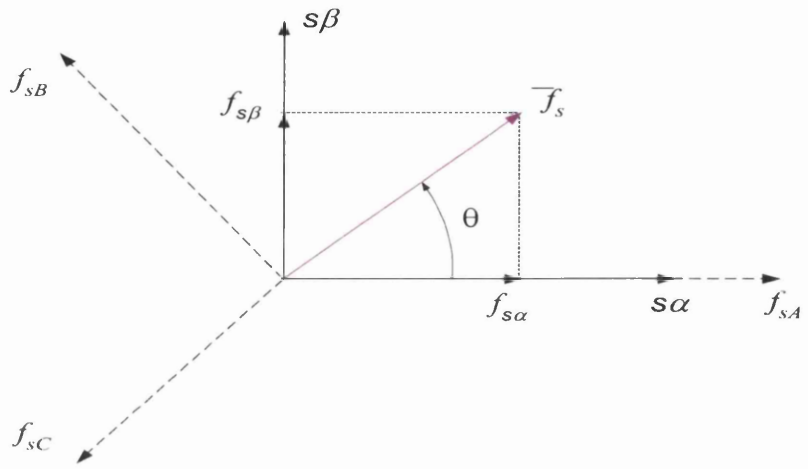


Figure 3-4: Projections of the stator space vectors.

Let f_s represent a general stator variable such as stator voltages, currents and flux linkages, then the complex space vector (also known as space phasor) in the stationary reference frame fixed to the stator is defined as

$$\begin{aligned}\overline{f_s^s} &= f_{s\alpha}^s + j\, f_{s\beta}^s \\ &= k\left(f_{sA} + a\, f_{sB} + a^2\, f_{sC}\right)\end{aligned} \tag{3.7}$$

where, $f^s{}_{s\alpha}$ and $f^s{}_{s\beta}$ represent the real and imaginary stator components in the stationary reference frame fixed to the stator, respectively, as shown in Figure 3-4. Furthermore, a and a^2 are spatial operators used to represent the phase shift of the stator winding, where $a = e^{j2\pi/3}$ and $a^2 = e^{j4\pi/3}$. In symmetrical three phase induction machine, the real and imaginary stator components $f_{s\alpha}$, $f_{s\beta}$ are fictitious, which are related to the three phase stator quantities (f_{sA}, f_{sB} and f_{sC}) as follows

$$f_{s\alpha}^{s} = k\left(f_{sA} - \frac{1}{2}f_{sB} - \frac{1}{2}f_{sC}\right) \tag{3.8}$$

and

$$f_{s\beta}^{s} = k\frac{\sqrt{3}}{2}\left(f_{sB} - f_{sC}\right) \tag{3.9}$$

where constant k is used as a scaling factor. For the non-power invariant form of the phase transformation from three phase to two phase components, k is equal to 2/3 (Vas 1990). It follows from the general equation (3.7) that the space vector does not contain the zero sequence component because the neutral point of the three phase system is assumed to be isolated. Thus if there is a zero sequence component an additional term must be added to the general equation. In other words, equation (3.7) is valid for any stator quantity as long as

$$f_{s0} = f_{sA} + f_{sB} + f_{sC} = 0 \tag{3.10}$$

where f_{s0} is the value of the stator zero sequence component. Similar to the stator general equation (3.7), rotor variables can be defined by a complex space vector in rotating reference frame fixed to the rotor as shown in Figure 3-5. Thus the general vector definition for any rotor variable is

$$\begin{aligned}\overline{f_r^r} &= f_{r\alpha}^{r} + j\, f_{r\beta}^{r} \\ &= k\left(f_{ra} + a\, f_{rb} + a^2\, \mathrm{f_{rc}}\right)\end{aligned} \tag{3.11}$$

where, $f^{r}{}_{r\alpha}$ and $f^{r}{}_{r\beta}$ represent the real and imaginary rotor components in the rotating reference frame fixed to the rotor, respectively. These two components are related to the three phase rotor quantities by

$$f_{r\alpha}^{r} = k\left(f_{ra} - \frac{1}{2}f_{rb} - \frac{1}{2}f_{rc}\right) \tag{3.12}$$

and

$$f_{r\beta}^{r} = k\frac{\sqrt{3}}{2}\left(f_{rb} - f_{rc}\right) \tag{3.13}$$

where constant k has the value of 2/3 for the non-power invariant form. It should be noted that it is also feasible to express the rotor variables in the stationary reference frame fixed to the stator. This can be achieved by multiplying the rotor frame vector presented in equation (3.11) by $e^{j\theta_r}$ and therefore it is defined as,

$$\overline{f_r^s} = \overline{f_r^r} e^{j\theta_r} \tag{3.14}$$

where $\overline{f_r^s}$ and $\overline{f_r^r}$ represent the rotor variables space vectors expressed in stationary reference frame fixed to the stator and rotating reference frame fixed to the rotor, respectively. Furthermore, it is also possible to express the stator variables in the rotational reference frame fixed to the rotor. This can be achieved by multiplying the stator frame vector presented in equation (3.7) by $e^{-j\theta_r}$ which yields

$$\overline{f_s^r} = \overline{f_s^s} e^{-j\theta_r} \tag{3.15}$$

The phase angle θ_r represents the angular displacement of the rotational reference frame with respect to the real axis ($f_{s\alpha}$) of the stator reference frame as shown in Figure 3-5.

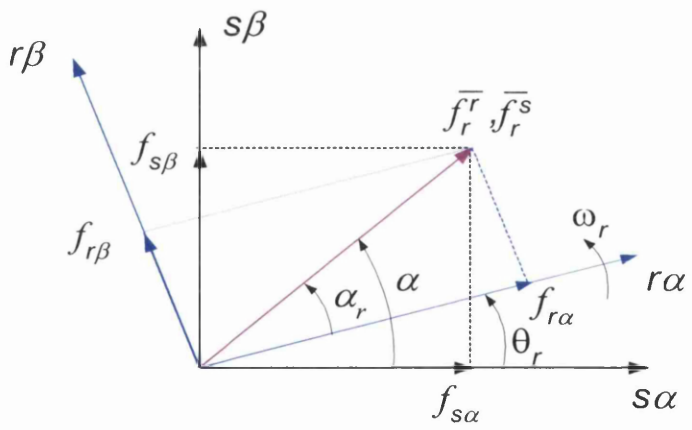


Figure 3-5: The relationship between the stationary and rotating reference frames.

3.3.1 Dynamic equations in the stationary reference frame:

According to the general equation (3.7), the three phase stator currents can be transformed to a space vector, whose real and imaginary parts are equal to the two stator current components denoted by $i^s_{s\alpha}$ and $i^s_{s\beta}$, respectively. Hence, the stator current vector in the stationary reference frame fixed to the stator can be expressed as,

$$\overline{i^s_s} = i^s_{s\alpha} + j\, i^s_{s\beta} \tag{3.16}$$

Similarly, it is possible to give the definition of the three phase stator flux linkages into their two axis components denoted by $\psi^s_{s\alpha}$ and $\psi^s_{s\beta}$ as,

$$\overline{\psi^s_s} = \psi^s_{s\alpha} + j\, \psi^s_{s\beta} \tag{3.17}$$

In equation (3.17), the real flux linkage component $\psi^s_{s\alpha}$ is defined by,

$$\psi^s_{s\alpha} = L_s i^s_{s\alpha} + L_m\left(i^r_{r\alpha}\cos\theta_r - i^r_{r\beta}\sin\theta_r\right) \tag{3.18}$$

and the imaginary flux linkage component $\psi^s_{s\beta}$ is given by,

$$\psi^s_{s\beta} = L_s i^s_{s\beta} + L_m\left(i^r_{r\alpha}\sin\theta_r + i^r_{r\beta}\cos\theta_r\right) \tag{3.19}$$

where L_s is the per-phase stator self inductance, while L_m represents the per-phase mutual (magnetizing) inductance between the stator and rotor windings. The current quantities $i^r_{r\alpha}$ and $i^r_{r\beta}$ in the equations above (3.18 and 3.19) are the two axis components of the rotor currents in the rotating reference frame fixed to the rotor. Substitution of equations (3.18) and (3.19) into equation (3.17) yields the following space vector for the stator flux linkage

$$\overline{\psi^s_s} = L_s\overline{i^s_s} + L_m\left(\overline{i^r_r}e^{j\theta_r}\right) \tag{3.20}$$

It follows from equation (3.20) that the stator flux linkage is made of two flux linkage space vector components. The first component ($L_s\overline{i_s^s}$) is the self flux linkage space vector of the stator phases, which is caused by the stator currents. The second component ($L_m\overline{i_r^r}e^{j\theta_r}$) is the mutual flux linkage space vector, which is caused by the rotor currents and expressed in the rotating reference frame. However, if the general transform equation (3.14) is considered, equation (3.20) would be,

$$\overline{\psi_s^s} = L_s\overline{i_s^s} + L_m\overline{i_r^s} \tag{3.21}$$

where $\overline{i_r^s}$ represents the rotor current space vector expressed in the stator reference frame fixed to the stator. Similar developments to the stator current and flux linkage space vectors can be held for the rotor currents and flux linkages. Thus according to the general equation (3.11), the three phase rotor currents can be transformed to a space vector, whose real and imaginary parts are equal to the two rotor current components denoted by $i_{r\alpha}^r$ and $i_{r\beta}^r$, respectively. Hence, the rotor current vector in the rotational reference frame fixed to the rotor can be expressed as

$$\overline{i_r^r} = i_{r\alpha}^r + j\, i_{r\beta}^r \tag{3.22}$$

By using the transform equation (3.14), the rotor current vector defined in equation (3.22) can be expressed in the stationary reference frame fixed to the stator as

$$\overline{i_r^s} = \overline{i_r^r}e^{j\theta_r} \tag{3.23}$$

Similarly to the definition of the stator flux linkage presented in equation (3.17), the rotor flux linkage formulated in the rotational reference frame fixed to the rotor is

$$\overline{\psi_r^r} = \psi_{r\alpha}^r + j\,\psi_{r\beta}^r \tag{3.24}$$

where

$$\psi_{r\alpha}^r = L_r i_{r\alpha}^r + L_m\left(i_{s\alpha}^s \cos\theta_r + i_{s\beta}^s \sin\theta_r\right) \tag{3.25}$$

and,

$$\psi_{r\beta}^{r} = L_r i_{r\beta}^{r} + L_m\left(-i_{s\alpha}^{s}\ sin\theta_r + i_{s\beta}^{s} cos\theta_r\right) \tag{3.26}$$

Substitution of equations (3.25) and (3.26) into equation (3.24) while considering the transform equation (3.15) yields the following space vector for the rotor flux linkage,

$$\begin{aligned}\overline{\psi_r^r} &= L_r \overline{i_r^r} + L_m\left(\overline{i_s^s} e^{-j\theta_r}\right) \\ &= L_r \overline{i_r^r} + L_m \overline{i_s^r}\end{aligned} \tag{3.27}$$

where $\overline{i_s^r}$ represents the stator current space vector expressed in the rotational reference frame fixed to the rotor. On the other hand, the production of an electromagnetic torque in an induction motor can be governed by the following equation

$$T_d = -\frac{3}{2} P \frac{L_m}{L_s} \overline{\psi_s^s} \times \overline{i_r^s} \tag{3.28}$$

where P is pole pair ($P = 2$ for a four-pole motor) . Finally, the space vectors of the stator and rotor voltages are formulated to complete the electrical dynamic model of the induction motor. The three-phase stator and short-circuited rotor (squirrel-cage type) voltage equations can be combined, respectively, in a single vectorial equation. Thus the stator voltage vector expressed in the stationary reference frame fixed to the stator is

$$\overline{v_s^s} = R_s \overline{i_s^s} + \frac{d}{dt} \overline{\psi_s^s} \tag{3.29}$$

Substitution of equation (3.21) into (3.29) and resolution of the current vectors into their two axis components yields the following two real stator voltage equations in the stationary reference frame

$$v_{s\alpha}^{s} = R_s i_{s\alpha}^{s} + L_s \frac{di_{s\alpha}^{s}}{dt} + L_m \frac{di_{r\alpha}^{s}}{dt} \tag{3.30}$$

$$v_{s\beta}^{s} = R_s i_{s\beta}^{s} + L_s \frac{di_{s\beta}^{s}}{dt} + L_m \frac{di_{r\beta}^{s}}{dt} \tag{3.31}$$

On the other hand, similar definitions hold for the rotor voltage vector expressed in the rotational reference frame fixed to the rotor as,

$$\overline{v_r^r} = R_r \overline{i_r^r} + \frac{d}{dt}\overline{\psi_r^r} \tag{3.32}$$

Substitution of equation (3.27) into (3.32) gives,

$$\overline{v_r^r} = R_r \overline{i_r^r} + L_r \frac{d\overline{i_r^r}}{dt} + L_m \frac{d\overline{i_s^s}}{dt} e^{-j\theta_r} \tag{3.33}$$

However, it is more convenient to express the stator and rotor variables in one reference frame instead of being formulated in two different frames. Here, the stationary reference frame fixed to the stator has been used as a common frame to define the motor variables. Hence, a complete dynamic model of an induction motor in the stationary coordinates can be constructed. Thus, the two real stator voltage equations presented in (3.30) and (3.31) are kept unchanged because they are naturally expressed in the stationary reference frame. On the other hand, the rotor voltages and currents ought to be transformed from the rotor reference frame to the stationary reference frame fixed to the stator. This can be achieved by applying the transform equations (3.14) and (3.15) on the rotor variables. Thus the rotor voltage equation presented in (3.33) can be expressed in the stationary reference frame as

$$\begin{aligned}\overline{v_r^s} e^{-j\theta_r} &= R_r \overline{i_r^s} e^{-j\theta_r} + L_r \frac{d\overline{i_r^s}}{dt} e^{-j\theta_r} + L_m \frac{d\overline{i_s^s}}{dt} e^{-j\theta_r} \\ &= R_r \overline{i_r^s} e^{-j\theta_r} + L_r e^{-j\theta_r} \frac{d\overline{i_r^s}}{dt} + L_r \overline{i_r^s}(-j) e^{-j\theta_r} \frac{d\theta_r}{dt} + L_m e^{-j\theta_r} \frac{d\overline{i_s^s}}{dt} + L_m \overline{i_s^s}(-j) e^{-j\theta_r} \frac{d\theta_r}{dt}\end{aligned} \tag{3.34}$$

In this equation, the complex term $e^{-j\theta_r}$ would cancel out and the derivative term $d\theta_r/dt$ represents the speed of the rotational reference frame (ω_r), where θ_r is the rotor angle (shown in Figure 3-5). Thus equation (3.34) becomes

$$\overline{v_r^s} = R_r \overline{i_r^s} + L_r \frac{d\overline{i_r^s}}{dt} - jL_r \overline{i_r^s}\omega_r + L_m \frac{d\overline{i_s^s}}{dt} - jL_m \overline{i_s^s}\omega_r \tag{3.35}$$

By resolving equation (3.35) into its two axis components results in the following two real rotor voltage equations in the stationary reference frame

$$v_{r\alpha}^s = R_r i_{r\alpha}^s + L_r \frac{di_{r\alpha}^s}{dt} + L_r i_{r\beta}^s \omega_r + L_m \frac{di_{s\alpha}^s}{dt} + L_m i_{s\beta}^s \omega_r \tag{3.36}$$

$$v_{r\beta}^s = R_r i_{r\beta}^s + L_r \frac{di_{r\beta}^s}{dt} - L_r i_{r\alpha}^s \omega_r + L_m \frac{di_{s\beta}^s}{dt} - L_m i_{s\alpha}^s \omega_r \tag{3.37}$$

From the stator and rotor voltage equations presented in (3.30), (3.31), (3.36) and (3.37), a complete electrical dynamic model expressed in the stator reference frame fixed to the stator is constructed and it can be simply presented in a matrix form as

$$\begin{bmatrix} v_{s\alpha}^s \\ v_{s\beta}^s \\ v_{r\alpha}^s \\ v_{r\beta}^s \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 \\ 0 & \omega_r L_m & R_r & \omega_r L_r \\ -\omega_r L_m & 0 & -\omega_r L_r & R_r \end{bmatrix} \begin{bmatrix} i_{s\alpha}^s \\ i_{s\beta}^s \\ i_{r\alpha}^s \\ i_{r\beta}^s \end{bmatrix} + \begin{bmatrix} L_s\rho & 0 & L_m\rho & 0 \\ 0 & L_s\rho & 0 & L_m\rho \\ L_m\rho & 0 & L_r\rho & 0 \\ 0 & L_m\rho & 0 & L_r\rho \end{bmatrix} \begin{bmatrix} i_{s\alpha}^s \\ i_{s\beta}^s \\ i_{r\alpha}^s \\ i_{r\beta}^s \end{bmatrix} \tag{3.38}$$

where ρ is the derivative operator (d/dt). However, to fully assess the performance of the induction motor under transient condition, the electrical model exhibited in a single matrix equation (3.38) is used together with the motion equation

$$\frac{d\omega_r}{dt} = \frac{1}{J}(T_d - t_l - B\omega_r) \tag{3.39}$$

where T_d is the produced electromagnetic torque and it is given in Newton meters (N.m), t_l is the external torque (N.m). J represents the inertia of the rotor and it is usually given as kg.m^2, B is the friction constant that represents mechanical losses due to windage and friction given in N.m/rad/sec, and ω_r is the rotor speed given in radians per second (rad/sec). Thus the matrix equation (3.39) combined with the motion equation presented in (3.39) and the torque equation (3.28) form a complete mathematical model of a symmetrical induction motor in the stationary reference frame fixed to the stator.

3.3.2 Dynamic equations in the synchronous reference frame:

In theory, the synchronous reference frame is a general reference frame with the direct and quadrature axes (d-q) rotating at synchronous speed ω_s. This indicates that each machine quantity (voltages, currents, fluxes etc.) for both stator and rotor can be expressed vectorially in the new reference frame in terms of their real and imaginary components (Murphy and Turnbull 1988; Vas 1990).

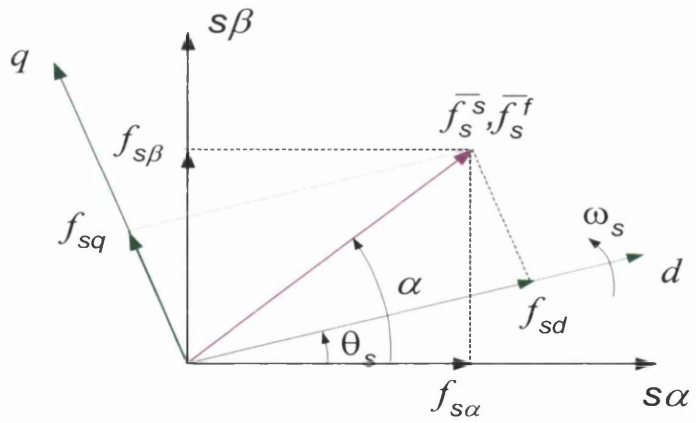


Figure 3-6: The relationship between the stationary and synchronous reference frames.

From the analysis presented earlier, Equations (3.14) and (3.15) in particular with the aid of Figure 3-5, the relationship between the stationary and rotational reference frames has been demonstrated. It has been shown that the space vector of any stator variable can be expressed in the rotational reference frame, and similarly the space vector of any rotor variable can be expressed in the stationary reference

frame. However, in the present analysis, it will be possible to express the stator and rotor variables in the synchronous frame denoted by d and q, which are oriented 90° to each other and rotating at synchronous speed ω_s.

Figure 3-6 shows the relationship between the stationary and synchronous reference frames, in which the direct axis (d) of the synchronous frame is displaced by an angle of θ_s from the real axis ($s\alpha$) of the stationary reference frame fixed to the stator. Thus the general stator equation expressed in the synchronous reference frame is

$$\begin{aligned} \overline{f_s^f} &= \overline{f_s^s} e^{-j\theta_s} \\ &= f_{sd}^f + j f_{sq}^f \end{aligned} \tag{3.40}$$

It follows from the general equation (3.40), that the space vectors of the stator voltages, currents and flux linkages can be formulated in a new reference frame rotating at synchronous speed as

$$\begin{aligned} \overline{v_s^f} &= \overline{v_s^s} e^{-j\theta_s} \\ &= v_{sd}^f + j v_{sq}^f \end{aligned} \tag{3.41}$$

$$\begin{aligned} \overline{i_s^f} &= \overline{i_s^s} e^{-j\theta_s} \\ &= i_{sd}^f + j\, i_{sq}^f \end{aligned} \tag{3.42}$$

$$\begin{aligned} \overline{\psi_s^f} &= \overline{\psi_s^s} e^{-j\theta_s} \\ &= \psi_{sd}^f + j\, \psi_{sq}^f \end{aligned} \tag{3.43}$$

Similar derivations hold for the space vectors of all the rotor variables to be expressed in the synchronous reference frame. Figure 3-7 illustrates the relationship between the rotor and the synchronous reference frame, in which the angular displacement between the real-axis of two frames is equal to θ_s - θ_r. Hence the general rotor equation expressed in the synchronous reference frame is,

$$\begin{aligned}\overline{f_r^f} &= \overline{f_r^r} e^{-j(\theta_s-\theta_r)} \\ &= f_{rd}^f + j f_{rq}^f\end{aligned} \tag{3.44}$$

This general equation is valid for rotor voltages, currents and flux linkages. They can be expressed in the synchronous reference frame as,

$$\begin{aligned}\overline{v_r^f} &= \overline{v_r^r} e^{-j(\theta_s-\theta_r)} \\ &= v_{rd}^f + j v_{rq}^f\end{aligned} \tag{3.45}$$

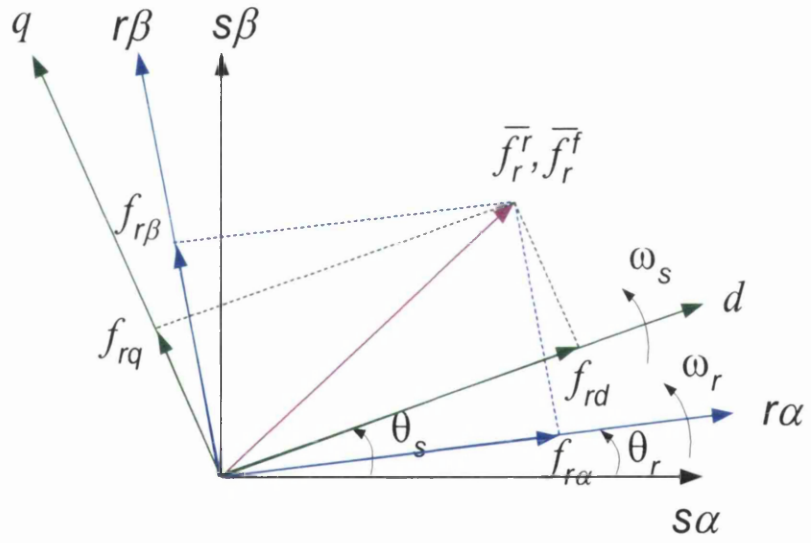


Figure 3-7: The relationship between the rotational and synchronous reference frames.

$$\begin{aligned}\overline{i_r^f} &= \overline{i_r^r} e^{-j(\theta_s-\theta_r)} \\ &= i_{rd}^f + j\, i_{rq}^f\end{aligned} \tag{3.46}$$

$$\begin{aligned}\overline{\psi_r^f} &= \overline{\psi_r^r} e^{-j(\theta_s-\theta_r)} \\ &= \psi_{rd}^f + j\, \psi_{rq}^f\end{aligned} \tag{3.47}$$

At this point, the dynamic equations that describe the transient behavior of the induction motor can be formulated in the synchronous reference frame. This can be achieved by adopting the following derivations. Recalling back the space vector (equation (3.29)) of the stator voltage, which is expressed in the stator reference frame fixed to the stator as

$$\overline{v_s^s} = R_s \overline{i_s^s} + \frac{d}{dt}\overline{\psi_s^s} \tag{3.48}$$

Substitution of the transform equations (3.41), (3.42) and (3.43) into (3.48), yields the following space vector of the stator voltage derived in the synchronous reference frame,

$$\begin{aligned} \overline{v_s^f} e^{j\theta_s} &= R_s \overline{i_s^f} e^{j\theta_s} + \frac{d}{dt}\overline{\psi_s^f} e^{j\theta_s} \\ &= R_s \overline{i_s^f} e^{j\theta_s} + j e^{j\theta_s} \frac{d\theta_s}{dt}\overline{\psi_s^f} + e^{j\theta_s} \frac{d}{dt}\overline{\psi_s^f} \end{aligned} \tag{3.49}$$

In this equation, the complex term $e^{j\theta_s}$ would cancel out and the derivative term $d\theta_s/dt$ represents the revolving speed of the synchronous reference frame (ω_s), where θ_s is the angle between the direct-axis (d) of the synchronous reference frame and the real-axis ($s\alpha$) of the stator reference frame (also shown in Figure 3-7). Thus equation (3.49) becomes

$$\overline{v_s^f} = R_s \overline{i_s^f} + j\omega_s \overline{\psi_s^f} + \frac{d}{dt}\overline{\psi_s^f} \tag{3.50}$$

where $\overline{i_s^f}$ and $\overline{\psi_s^f}$ are the space vectors of the stator currents and stator flux linkages, respectively, in the synchronous reference frame. It can be observed by examining equation (3.50) that a new voltage term ($j\omega_s \overline{\psi_s^f}$) appeared in the space vector of the stator voltage compared to that in equation (3.48). The space vector of the stator flux linkage in the synchronous frame is given by

$$\overline{\psi_s^f} = L_s \overline{i_s^f} + L_m \overline{i_r^f} \tag{3.51}$$

Substitution of equation (3.51) into (3.50) and resolution of the current vectors into their two axis components yields the following two real stator voltage equations in the synchronous reference frame

$$v_{sd}^{f} = R_s i_{sd}^{f} + L_s \frac{di_{sd}^{f}}{dt} + L_m \frac{di_{rd}^{f}}{dt} - \omega_s L_s i_{sq}^{f} - \omega_s L_m i_{rq}^{f} \tag{3.52}$$

$$v_{sq}^{f} = R_s i_{sq}^{f} + L_s \frac{di_{sq}^{f}}{dt} + L_m \frac{di_{rq}^{f}}{dt} + \omega_s L_s i_{sd}^{f} + \omega_s L_m i_{rd}^{f} \tag{3.53}$$

On the other hand, similar definitions hold for the rotor voltages. Recalling back the space vector of the rotor voltages in the rotational reference frame presented in equation (3.32) as

$$\overline{v_r^r} = R_r \overline{i_r^r} + \frac{d}{dt}\overline{\psi_r^r} \tag{3.54}$$

Substitution of the transform equations (3.45), (3.46) and (3.47) into (3.54), yields the following space vector of the rotor voltage derived in the synchronous reference frame

$$\begin{aligned} \overline{v_r^f} e^{j(\theta_s-\theta_r)} &= R_r \overline{i_r^f} e^{j(\theta_s-\theta_r)} + \frac{d}{dt}\overline{\psi_r^f} e^{j(\theta_s-\theta_r)} \\ &= R_r \overline{i_r^f} e^{j(\theta_s-\theta_r)} + j e^{j(\theta_s-\theta_r)} \frac{d(\theta_s-\theta_r)}{dt}\overline{\psi_r^f} + e^{j(\theta_s-\theta_r)} \frac{d}{dt}\overline{\psi_r^f} \end{aligned} \tag{3.55}$$

In this equation, the complex term $e^{j(\theta_s-\theta_r)}$ would cancel out and the derivative term $d(\theta_s - \theta_r)/dt$ can be seen as the difference between the synchronous speed and the rotor speed (ω_{ms} - ω_{mr}) of the induction motor which equals to the slip frequency ω_{slip}. Thus equation (3.55) becomes

$$\overline{v_r^f} = R_r \overline{i_r^f} + j\omega_{slip}\overline{\psi_r^f} + \frac{d}{dt}\overline{\psi_r^f} \tag{3.56}$$

where $\overline{i_r^f}$ and $\overline{\psi_r^f}$ are the space vectors of the rotor currents and rotor flux linkages, respectively, in the synchronous reference frame rotating at synchronous speed ω_s. The space vector of the stator flux linkage in the synchronous frame is given by,

$$\overline{\psi_r^f} = L_r \overline{i_r^f} + L_m \overline{i_s^f} \tag{3.57}$$

Substitution of equation (3.57) into (3.56) and resolution of the current vectors into their two axis components results in the following two real rotor voltage equations in the synchronous reference frame

$$v_{rd}^f = R_r i_{rd}^f + L_r \frac{di_{rd}^f}{dt} + L_m \frac{di_{sd}^f}{dt} - \omega_{slip} L_r i_{rq}^f - \omega_{slip} L_m i_{sq}^f \tag{3.58}$$

$$v_{rq}^f = R_r i_{rq}^f + L_r \frac{di_{rq}^f}{dt} + L_m \frac{di_{sq}^f}{dt} + \omega_{slip} L_r i_{rd}^f + \omega_{slip} L_m i_{sd}^f \tag{3.59}$$

Thus from the stator and rotor voltage equations presented in (3.52), (3.53), (3.58) and (3.59), a complete electrical dynamic model expressed in the synchronous reference frame rotating at synchronous frequency ω_s is obtained and it can be simply presented in a matrix form as

$$\begin{bmatrix} v_{sd}^f \\ v_{sq}^f \\ v_{rd}^f \\ v_{rq}^f \end{bmatrix} = \begin{bmatrix} R_s & -\omega_s L_s & 0 & -\omega_s L_m \\ \omega_s L_s & R_s & \omega_s L_m & 0 \\ 0 & -\omega_{slip} L_m & R_r & -\omega_{slip} L_r \\ \omega_{slip} L_m & 0 & \omega_{slip} L_r & R_r \end{bmatrix} \begin{bmatrix} i_{sd}^f \\ i_{sq}^f \\ i_{rd}^f \\ i_{rq}^f \end{bmatrix} + \begin{bmatrix} L_s\rho & 0 & L_m\rho & 0 \\ 0 & L_s\rho & 0 & L_m\rho \\ L_m\rho & 0 & L_r\rho & 0 \\ 0 & L_m\rho & 0 & L_r\rho \end{bmatrix} \begin{bmatrix} i_{sd}^f \\ i_{sq}^f \\ i_{rd}^f \\ i_{rq}^f \end{bmatrix} \tag{3.60}$$

where ρ is the derivative operator (d/dt). If ω_s is set to zero in the above matrix equation, then the matrix equation (3.38) is obtained, which models the induction motor in the stationary reference frame fixed to the stator. The voltage equations presented in (3.60) are valid for transient and steady state operations for the induction motor. However, to fully assess its performance under transient conditions,

the following motion equation as in (3.39) is required together with the electrical model exhibited in the matrix equation (3.60)

$$\frac{d\omega_r}{dt} = \frac{1}{J}(T_d - t_l - B\omega_r) \tag{3.61}$$

where T_d is the developed electromagnetic torque of the induction motor and it can be expressed in terms of stator current and flux linkage space vectors in synchronous reference frame as

$$T_d = \frac{3}{2}P\left(\overline{\psi_s^f} \times \overline{i_s^f}\right) \tag{3.62}$$

Substitution of equation (3.51) into (3.62) yields,

$$\begin{aligned} T_d &= \frac{3}{2}P\left(\left[L_s\overline{i_s^f} + L_m\overline{i_r^f}\right] \times \overline{i_s^f}\right) \\ &= \frac{3}{2}P\,L_m\left(\overline{i_r^f} \times \overline{i_s^f}\right) \end{aligned} \tag{3.63}$$

where the vector product $\overline{i_s^f} \times \overline{i_s^f}$ gives zero. If the stator and rotor current space vectors are resolved into their two-axis components as defined in equations (3.42) and (3.46) which are expressed in the synchronous reference frame, then the torque equation in (3.63) becomes

$$T_d = \frac{3}{2}P\,L_m\left(i_{rd}^f i_{sq}^f - i_{rq}^f i_{sd}^f\right) \tag{3.64}$$

3.3.3 Computer simulation of induction motor model in synchronous reference frame:

The electrical model of the induction motor presented in matrix form as in (3.60) can be rearranged to be expressed in a state space equation to form the basis of system dynamics modeling. This can be achieved by first representing the matrix equation into a single algebraic equation as

$$U = ZX + L\dot{X} \tag{3.65}$$

where

$$U = \begin{bmatrix} v_{sd}^f \\ v_{sq}^f \\ v_{rd}^f \\ v_{rq}^f \end{bmatrix}, \quad Z = \begin{bmatrix} R_s & -\omega_s L_s & 0 & -\omega_s L_m \\ \omega_s L_s & R_s & \omega_s L_m & 0 \\ 0 & -\omega_{slip} L_m & R_r & -\omega_{slip} L_r \\ \omega_{slip} L_m & 0 & \omega_{slip} L_r & R_r \end{bmatrix}, \quad L = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix}$$

$$X = \begin{bmatrix} i_{sd}^f & i_{sq}^f & i_{rd}^f & i_{rq}^f \end{bmatrix}^T$$

Then, equation (3.65) can be rearranged to give

$$\dot{X} = AX + BU \tag{3.66}$$

where $A = -L^{-1}Z$ and $B = L^{-1}$

It follows from the motion equation (3.61) and the state space equation (3.66), that the system dynamics of an induction motor represent the solution of fifth order differential equation system. In other words, the inherent nonlinear characteristics of the induction motor give rise to a complex dynamic system. However, the use of modern numerical integration methods such as *Runge-Kutta* and *Euler* methods have made possible the solution of such complex system.

In the present work, computer simulations were carried out in SIMULINK and MATLAB to investigate the complex dynamics of the induction motor when it is subjected to operate at rated motor parameters SIMULINK (SIMULINK User's Guide 1992; Brian and Breiner 1995; Leonard and Levine 1995). Furthermore, the motor dynamics are examined when it is fed by two different voltage supplies, which are the ideal sinusoidal voltage supply and the PWM-based VSI inverter, respectively. Comparison studies between the simulation results obtained under these two modes of operations are also considered.

The induction motor is simulated using the dynamic equations that are formulated in the synchronous reference frame and rearranged into a state space form as presented in (3.66). Thus, the derivative of each stator and rotor currents can be integrated to determine their instantaneous values through step-by-step process of integration.

The fourth order Runge-Kutta method is employed in this simulation as a numerical integration technique to compute the *d-q* axis stator and rotor currents formulated in (3.66). Consequently, the stator and rotor flux linkages are calculated together with the electromagnetic torque by using equations (3.51), (3.57) and (3.64), respectively. The rotor speed is obtained from the integration of the motor acceleration according to the motion equation presented in (3.61). These procedures are then repeated for each step time taken as 0.5 msec for a complete simulation time of 1.2 sec.

3.3.3.1 Simulations of induction motor fed by an ideal sinusoidal voltage supply:

Here, the dynamic model of a 3-hp induction motor is simulated to illustrate its transient behavior upon direct on-line start when it is fed from an ideal three-phase voltage source. Each phase provides a pure sinusoidal voltage signal of $240V_{RMS}$ and nominal frequency of 50 Hz. The rated parameters of the simulated induction motor were determined experimentally and are shown in Table 3.1. In what follows, the transient characteristics of some of the motor variables during on-line starting with fixed supply voltage and constant frequency will be discussed.

	Stator	***Rotor***	***Magnetising Parameter***
3-hp, 410V, 50Hz *4 pole, 1415 r/min* *Friction constant = 0.0028 N.m/rad/sec* *Drive inertia = 0.055 kg.m²*	$R_s = 1.88\ \Omega$ $L_{ls} = 0.0125H$ $L_s = 0.1649H$	$R_r = 2.72\ \Omega$ $L_{lr} = 0.0125H$ $L_r = 0.1649H$	$L_m = 0.1524H$

Table 3.1: Parameters of induction motor.

Throughout the simulation, the load torque is set to 15.1 Nm. Figure 3-8 shows the profile of the rotor speed response during an on-line starting operation, where unmodulated ideal three-phase voltage signals are applied to the motor. The graph also shows that the rotor speed reaches its maximum value in less than 0.5 seconds at rated torque.

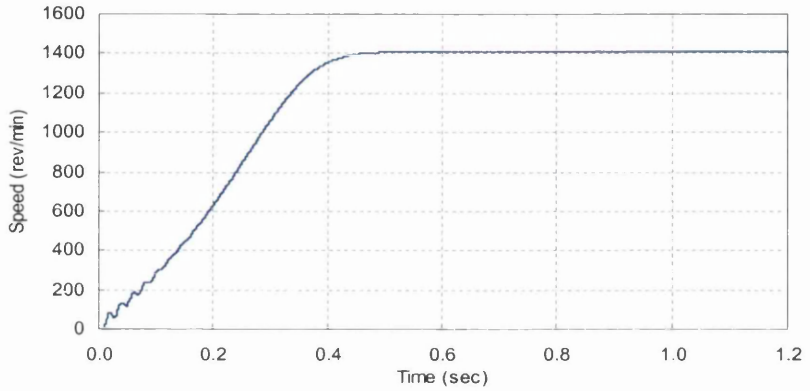


Figure 3-8: Rotor speed response of the induction motor at rated operation.

As the motor accelerates and the speed builds up, the electromagnetic torque oscillates (pulsates) dramatically for 0.2 seconds as shown in Figure 3-9. The torque curve then converges to its maximum breakdown value before it settles down to its rated value after 0.5 seconds.

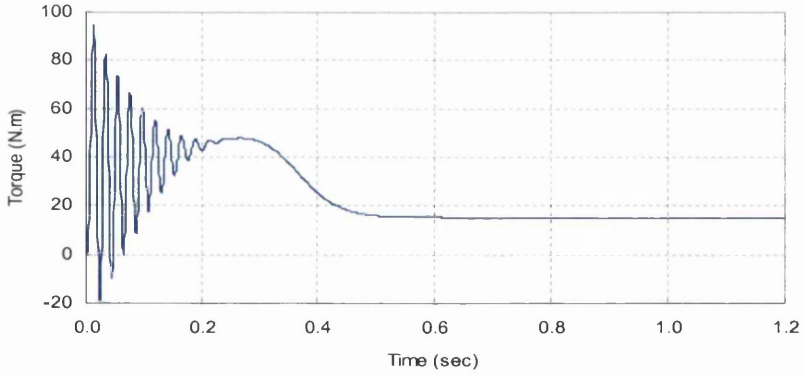


Figure 3-9: The developed electromagnetic torque of the induction motor.

The stator current upon direct on-line start operation is depicted in Figure 3-10. This figure shows that the final steady state stator current is equal to 9.2A peak, which corresponds to $6.5A_{RMS}$. However, the

rated stator current quoted on the nameplate of the motor is $5A_{RMS}$. Thus, the motor simulation returned an increase of $1.5A_{RMS}$ over the rated current value. Therefore, the no-load test and locked rotor test were repeated several times to confirm the values of the motor parameters given in Table 3.1. The results of these testes are shown in Appendix A.

The steady state equivalent circuit of the induction motor was then used to calculate the stator current from those parameters in order to confirm the simulation results. The calculated stator current was found to be equal to $6.43A_{RMS}$, which is approximately equal to the final stator current value returned by the simulation. This concludes that the final value of the stator current when the motor is running at full load is not $5A_{RMS}$ as quoted on the nameplate of the motor, which was manufactured in 1965. This at the same time means that the rated power of the motor (quoted as 3-hp on the nameplate) does not apply any more (no-load losses in a standard induction motor do not normally exceed 10% of the rated power; however no-load loss in the test was measured as 1030W, which is 46% of the rated power, as shown in Appendix A). This means that the rated torque value, applied in the study (15.1 Nm), is a notional one, since it was calculated using the 3-hp as the rated power.

Thus, giving the age of the motor of 36 years, it is almost certain that winding and insulation wear could be the reason behind the inequality between the stator current value returned by the simulation and that specified on the motor nameplate. Therefore, the value of $6.5A_{RMS}$ will be used hereinafter as the final stator current value for the induction motor at full load.

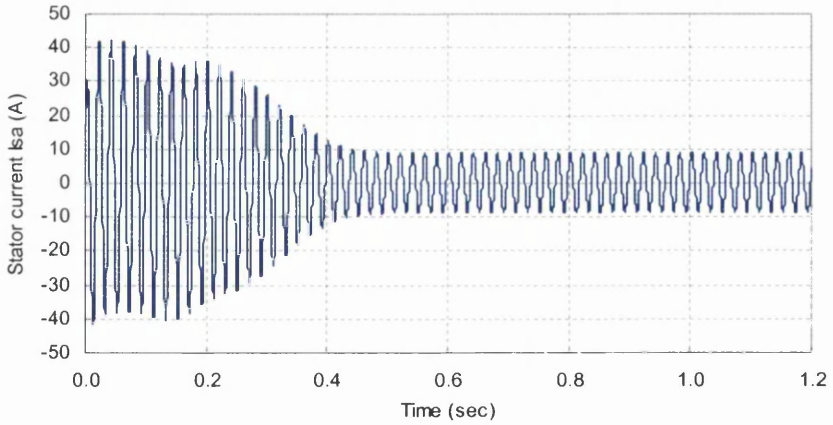


Figure 3-10: Stator current of the induction motor.

Figure 3-11 shows the profile of the two axis components for the stator and rotor currents, respectively. It can be observed that the inrush current for all cases remains oscillating for 0.2 seconds and it keeps decreasing until the steady state level is reached after 0.5 seconds. This would explain the transient behavior of the electromagnetic torque and the stator current variables as depicted earlier in Figures 3-9 and 3-10, respectively.

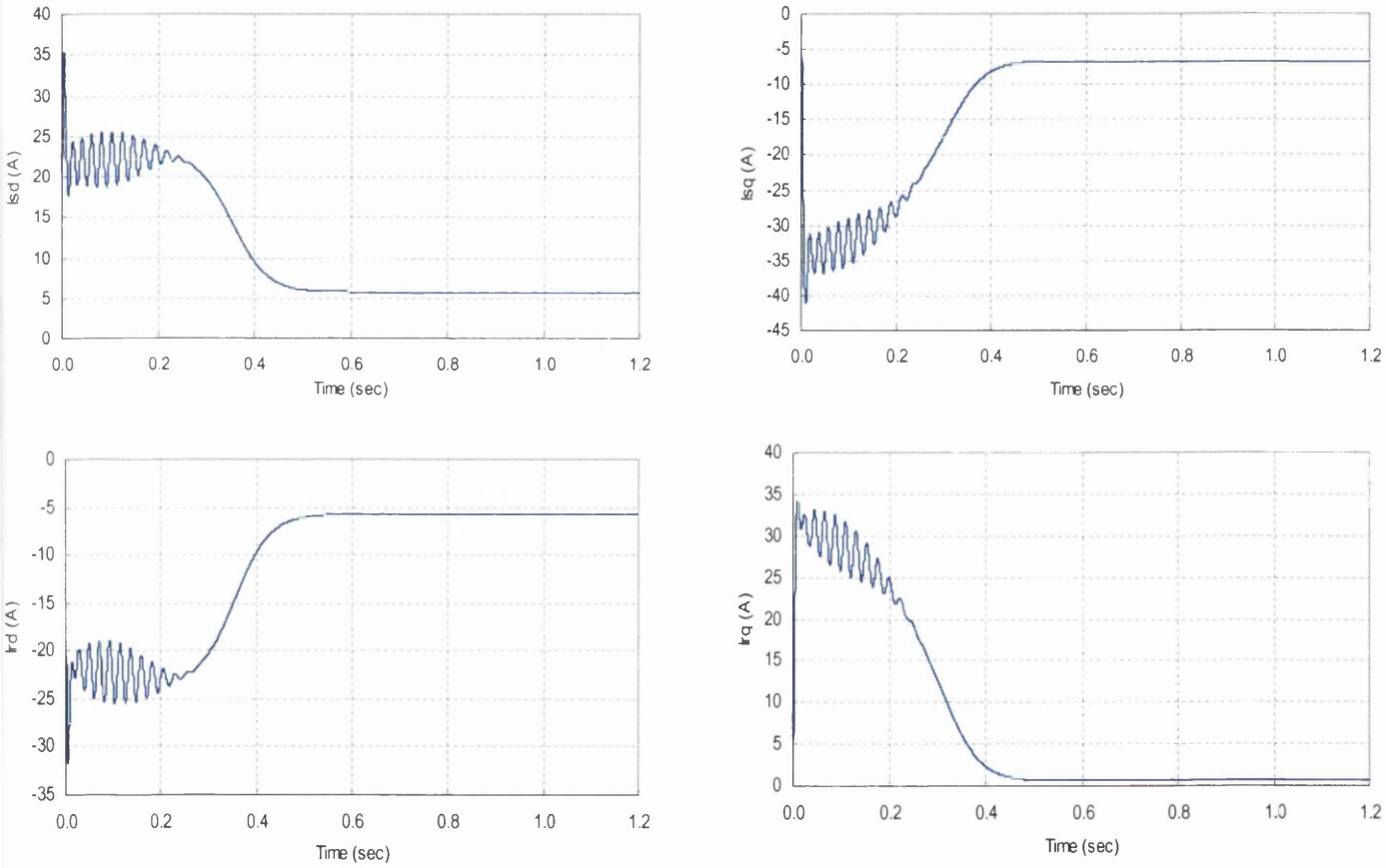


Figure 3-11: Stator and rotor two-axis current components (i_{sd} , i_{sq} , i_{rd} , i_{rq}).

The dynamic behavior of the stator flux linkage during starting operation is in Figure 3-12. The graph shows fluctuation for 0.2 seconds due to the high inrush current flowing in the stator windings during the starting stage. These oscillations then disappear immediately after the stator current starts uniformly decreasing until it reaches its final steady state value. The d and q components of the stator flux are also shown in Figure 3-13.

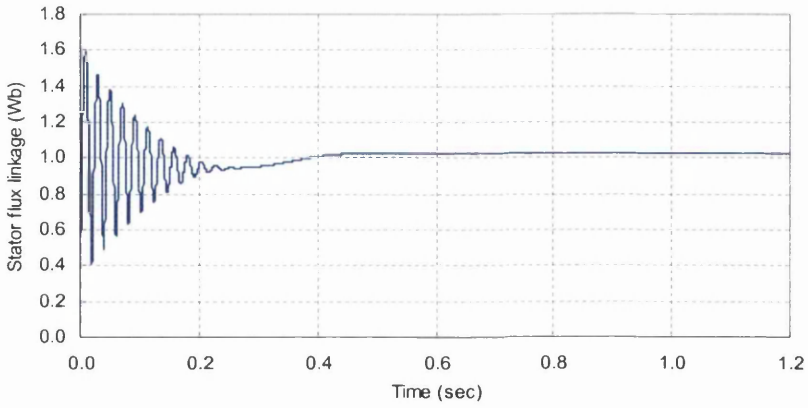
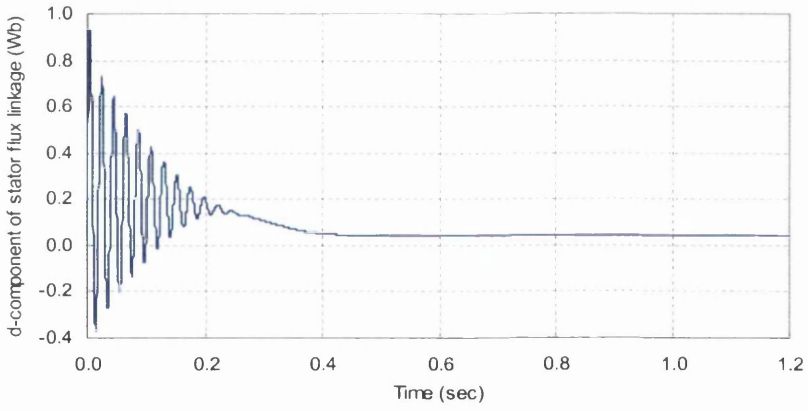


Figure 3-12: Stator flux linkage of the induction motor



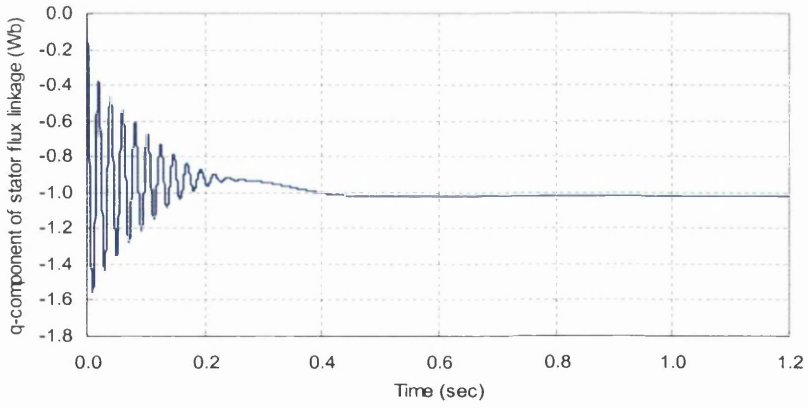


Figure 3-13: d and q components of stator flux linkage.

The resulting rotor flux linkage is shown in Figure 3-14. The graph shows that the rotor flux level oscillates for a short period of 0.2 seconds. It can be seen that the growth takes approximately 0.4 seconds, which is 6.5 times the rotor time constant (0.061 sec). In addition to the electrical variables of the induction motor, the transient performance of the shaft power is shown in Figure 3-15.

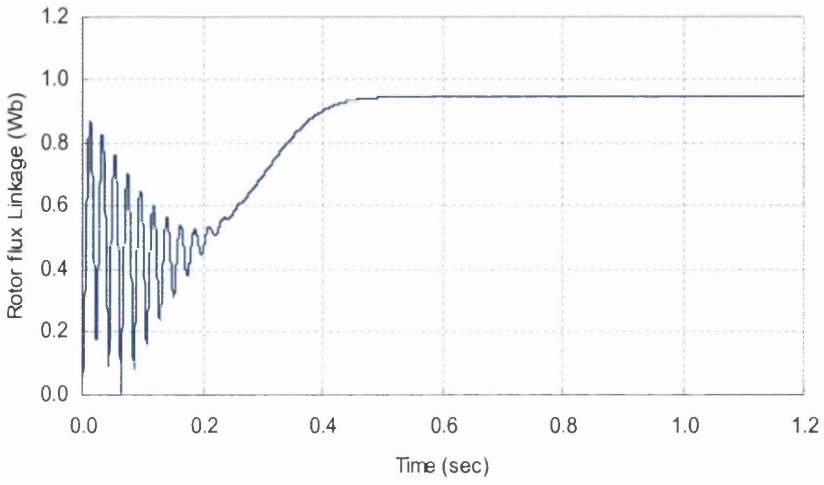


Figure 3-14: Magnitude of the rotor flux linkage of the induction motor.

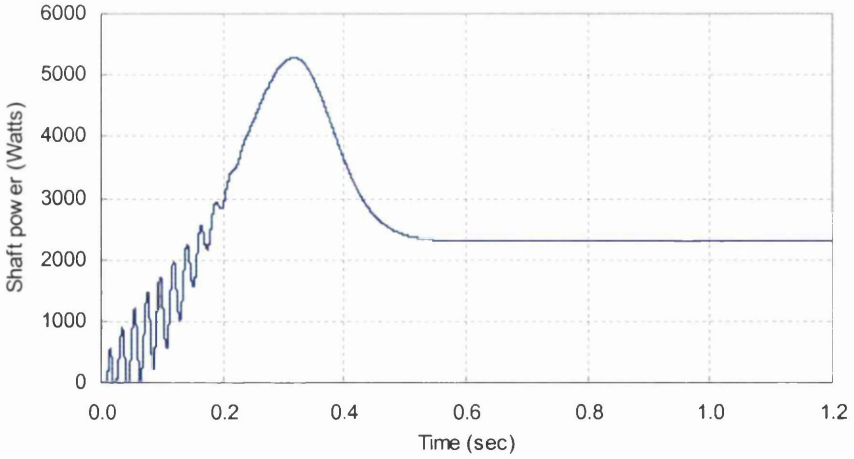


Figure 3-15: shaft power of the induction motor.

3.3.3.2 *Simulations of induction motor fed by PWM voltage waveforms:*

In this scheme, the transient and steady state performances of the induction motor are investigated during direct on-line starting operation when it is supplied by three-phase PWM voltage waveforms. The two-level *Asymmetric Regular Sampled SPWM* technique discussed in Chapter 2 is employed in the present simulation studies. The DC link voltage of the inverter is assumed to be a perfect DC voltage source and the switching power devices are assumed to be ideal. The simulated motor is the same as that used in the previous simulations.

The frequency ratio f_r is selected to be 9 and the modulation ratio M_r is set to 1, while the modulating frequency is f_m = 50 *Hz*. The DC-link voltage of the generated PWM signal is set to a smooth voltage value of 678V, because this generates a fundamental voltage with an RMS value of 240V (Murphy and Turnbull 1988; Mohan et al. 1995). Therefore, the results obtained would be comparable to the previous characteristic results of the induction motor. Figure 3-16 shows the generated per-phase PWM pole voltage in one electrical cycle with duration of 20 msec.

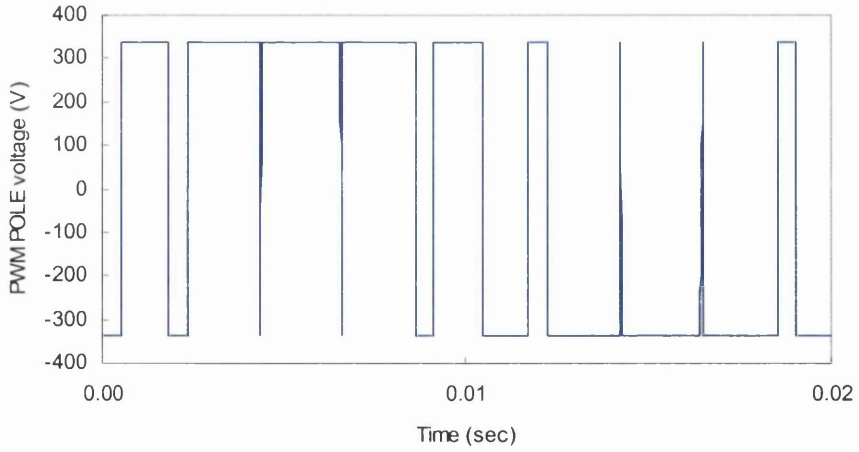


Figure 3-16: The generated PWM pole voltage.

It should be noted that the induction motor model accepts only phase to neutral voltage as a supply input. In other words, since the PWM signal generates pole voltage that swings between positive to negative levels, this voltage has to be transformed to a suitable phase to neutral voltage before it is fed to the induction motor model (Ho and Sen 1986). The required transformation is given by,

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} V_{ao} \\ V_{bo} \\ V_{co} \end{bmatrix} \tag{3.67}$$

where V_{an}, V_{bn} and V_{cn} are the three phase voltages, while V_{ao}, V_{bo} and V_{co} are the three PWM pole voltages. Figure 3-17 depicts the patterns of the phase voltages after the transform equation (3.67) has been applied on the PWM pole voltages, in order to drive the motor inputs.

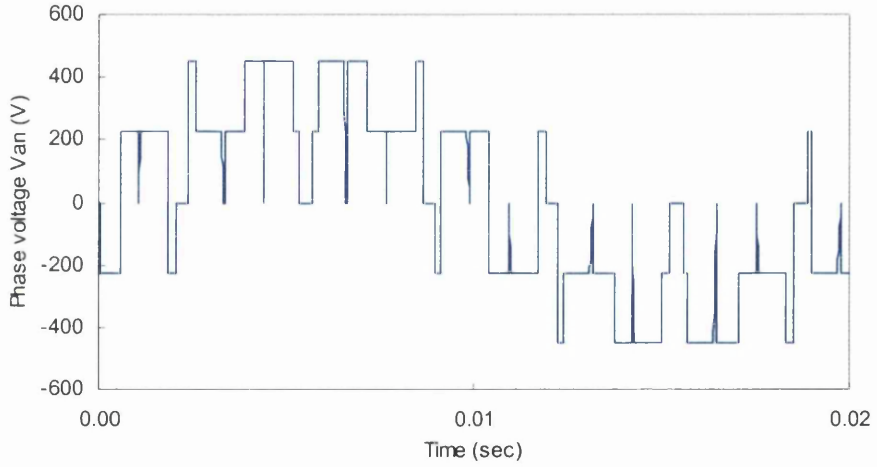


Figure 3-17: The generated phase voltage.

The phase voltage becomes a 5 level waveform, where the voltage is either 1/3 V_{dc}, 2/3 V_{dc}, 0, -1/3 V_{dc} or –2/3 V_{dc}. As a result of this voltage transformation, the motor model will be driven by three phase voltage waveforms whose shapes are similar to that of Figure 3-17, with a phase shift of 120° between each other. The shape of the resulted line voltage signal is depicted in Figure 3-18.

Inspection of this waveform shows that the voltage level changes between 0 and $+V_{dc}$ or between 0 and $-V_{dc}$. Furthermore, the number of pulses in each half cycle is equal to the frequency ratio f_r, which has been chosen to be 9. In what follows, the starting characteristics of the induction motor with PWM voltage operation will be discussed. Moreover, comparison between the induction motor results obtained here with previous results will be highlighted.

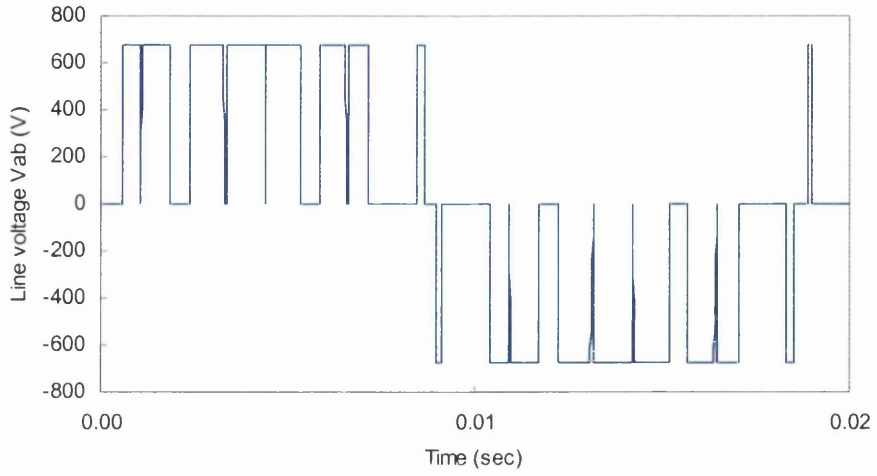


Figure 3-18: The generated line voltage.

Figure 3-19 shows the curve of the rotor speed response during an on-line starting operation, where three PWM voltage signals are fed to the motor. The graph shows an identical response to that obtained in the previous section. However, additional ripples appear in the speed curve at the steady state due to the switching nature of the PWM voltage signals. The effects of increasing the value of the frequency ratio f_r on the induction motor behavior will be considered.

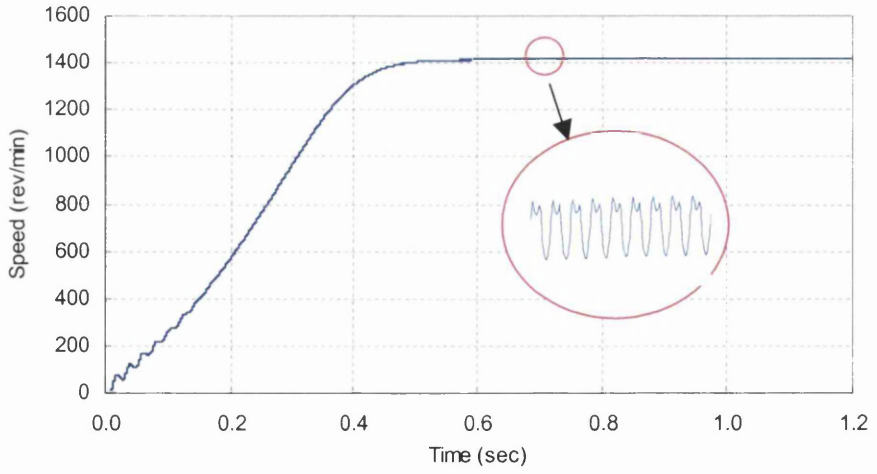


Figure 3-19: Rotor speed response of the induction motor with PWM operation.

As the motor accelerates and the speed builds up, the electromagnetic torque pulsates drastically for 0.2 seconds as shown in Figure 3-20. The torque curve then converges to its maximum breakdown value before it settles down to its rated value after 0.4 seconds. It can be observed that the torque signal during transient and steady state periods suffers from very high ripples, which is known as the torque pulsation.

It has been shown in literature (Bowes and Midoun 1988; Ho and Sen 1990), that the main cause of the speed oscillation and the torque pulsation is the existence of harmonics in the PWM-based inverter voltage waveforms. Although these effects can be neglected for higher speed operation, they may become very significant at low speed (Murphy and Turnbull 1988). These oscillation effects can be minimized by increasing the value of the frequency ratio f_r. This is always beneficial for the induction motor performance but it will be at the expense of increasing the switching losses of the power devices in the inverter circuit (Boost and Ziogas 1988; Rashid 1993).

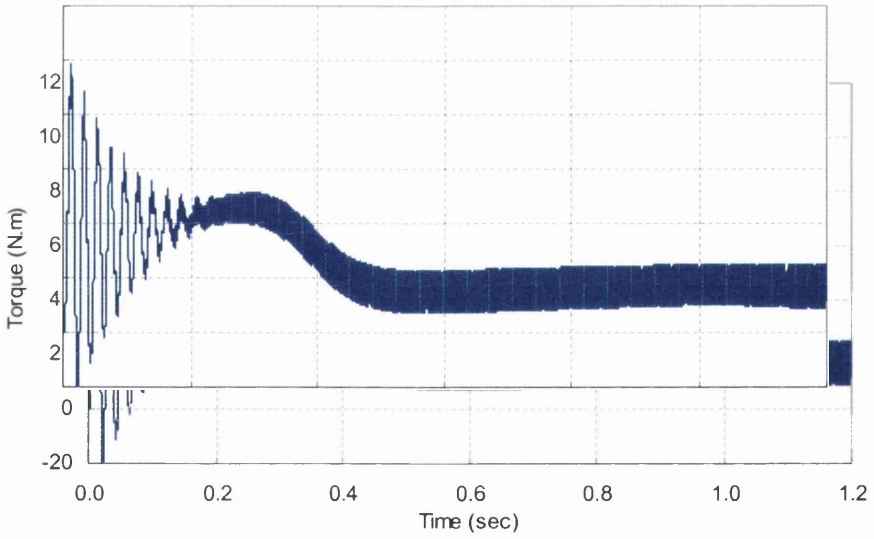


Figure 3-20: The developed electromagnetic torque of the induction motor with PWM operation

The resulted stator current upon direct on-line start with PWM operation is depicted in Figure 3-21. This graph shows close similarity to the stator current obtained in the previous section in terms of transient and steady state dynamic behavior. However, with the PWM operation, there are larger current peaks accompanied with ripples due to the insufficient low value of frequency ratio ($f_r = 9$). The transient behavior of the stator flux linkage is shown in Figure 3-22.

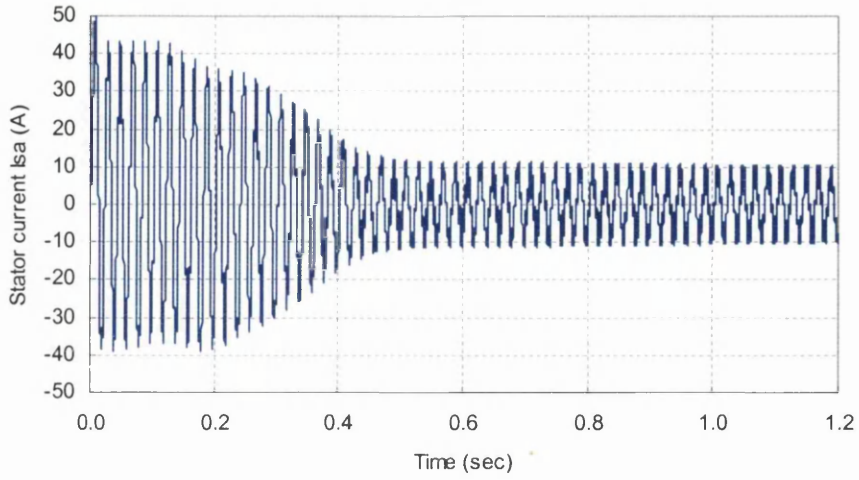


Figure 3-21: Stator phase current of the induction motor with PWM operation.

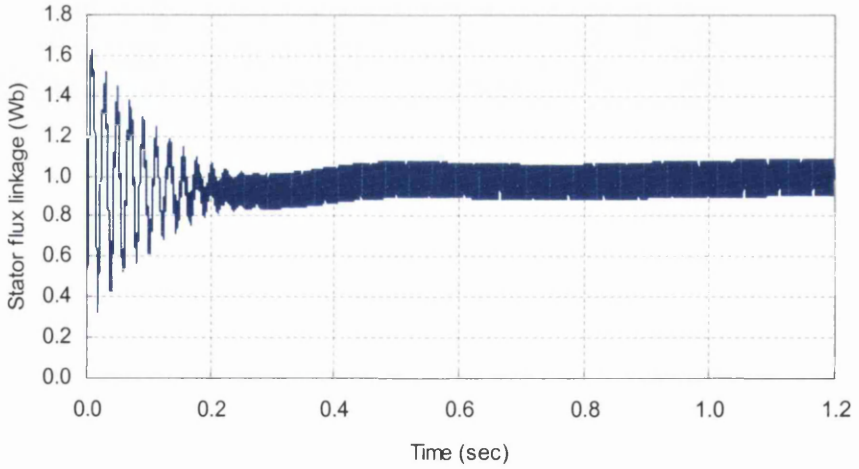


Figure 3-22: Stator flux linkage of the induction motor with PWM operation.

Figure 3-22 shows large flux fluctuation for 0.2 seconds due to the high inrush current flowing in the stator windings during the starting stage. Due to the PWM switching nature, high frequency ripples appear virtually in the stator flux linkage at steady state. The d and q components of the stator flux are also shown in Figure 3-23. The resultant rotor flux linkage is shown in Figure 3-24. The graph shows that the overall behavior of the rotor flux level approximately coincides with that obtained in the previous section. However, at the steady state, rotor flux ripples are present.

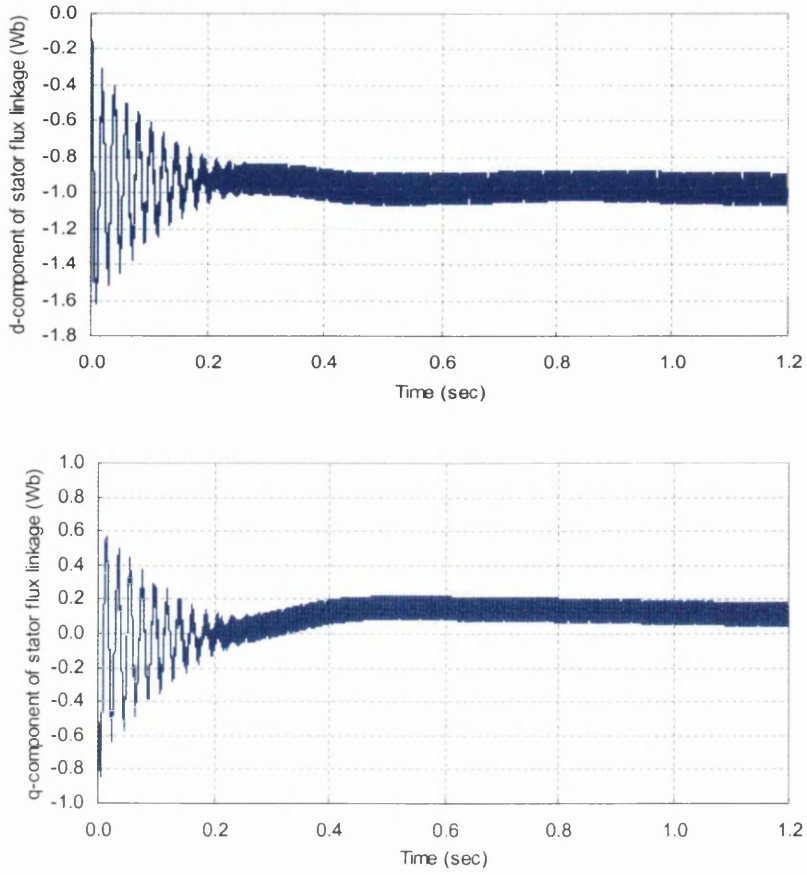


Figure 3-23: d and q components of stator flux linkage.

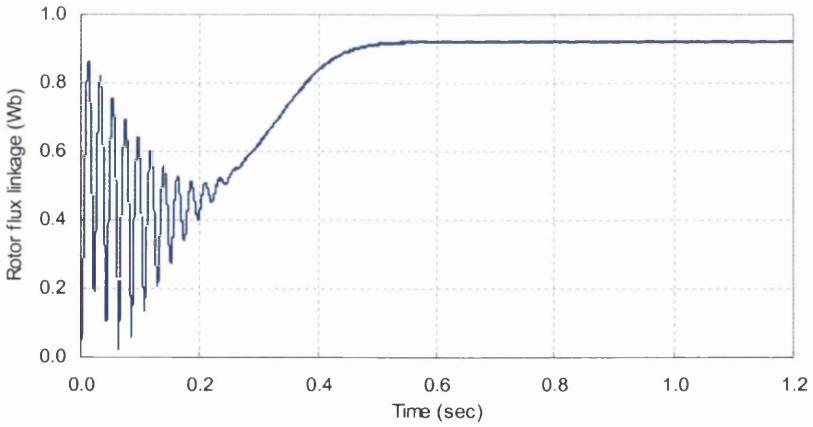


Figure 3-24: Magnitude of the rotor flux linkage of the induction motor with PWM operation.

In theory, the harmonics content in the output voltage and current waveforms are fairly reduced with the increase in the value of the frequency ratio. This causes the predominant harmonics to be pushed into a higher frequency range around the carrier frequency f_c, hence permitting smooth stator current signals and reduced torque pulsation. The transient effects on the motor current, and torque caused by the increase of the frequency ratio from 9 to 15 are shown in Figure 3-25. Inspection of these graphs shows that the generated ripple due to the harmonics content in the waveforms are reduced by increasing the value of the frequency ratio.

(a) *(b)*

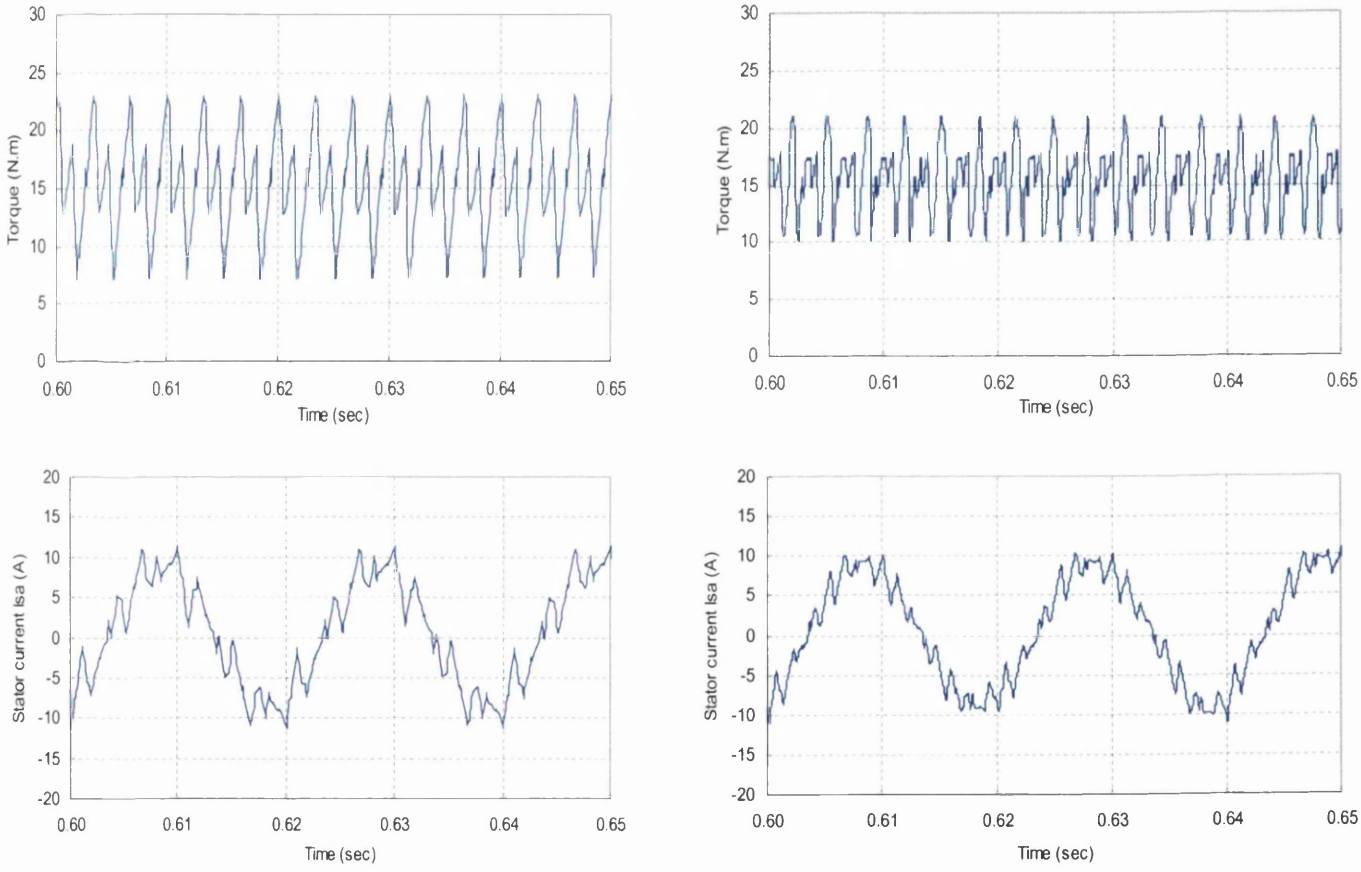


Figure 3-25: Profiles of torque and stator current at different frequency ratio: (a) f_r = 9, (b) f_r =15.

3.4 Summary:

In this chapter, the equivalent circuit and dynamic model for the induction motor have been developed for the steady state and transient response analysis, respectively. The space vectors of various motor quantities (stator and rotor *m.m.f.s*, stator and rotor flux linkages, etc.) have been reviewed. The derivation of the motor dynamic model in both stationary and d-q synchronously rotating reference frame has been given. Computer simulations of the motor dynamic model carried out in MATLAB and SIMULINK have been presented. The following are the main points made in this chapter:

1. The per-phase equivalent circuit of the induction motor is sufficient to highlight the essential features of the motor performance under steady state operation. An alternative model, termed dynamic model, is used to represent the dynamics of the induction motor and describe its real performance under both transient and steady state conditions.

2. There exist several dynamic models, which are based on the selection of desired reference frame. The most common reference frames are the stationary frame, the rotor frame, and the synchronous frame. The model equations expressed in the synchronous frame is the most commonly model for transient and steady state analysis. It also forms the basis for designing various forms of control of ac drives such as vector control. This model is adopted for computer simulation of scalar control in chapter 6.

3. One of the prime causes of speed oscillation and the torque pulsation in the induction motor is the existence of harmonics in the PWM-based source inverter. Although these effects can be neglected for higher speed operation, they may become very significant at low speed. Increasing the value of the frequency ratio of the PWM waveforms can minimize the oscillation effects. This is always beneficial for the induction motor performance but it will be at the expense of increasing the switching losses of the power devices in the inverter circuit.

Chapter 4, Application of fuzzy logic algorithms:

4.1 Introduction:

In recent years, fuzzy logic has emerged as one of the most attractive control algorithms used in several applications in the absence of complete and precise mathematical information. Its essence lies in being closer to the human thinking and natural language than the traditional control systems. It deals with objects in terms of degree of membership with all possible grades of logic between 0 and 1, and the shades of grey between white and black.

Fuzzy logic control is essentially based on the theory of fuzzy sets, which was introduced at first by Zadeh (1965) and extended to the developments of fuzzy relation concepts in 1972 (Chang and Zadeh 1972). An important breakthrough in this development was presented in 1973 (Zadeh 1973), which introduced the concept of linguistic variables and fuzzy conditional statements (fuzzy rules). A linguistic variable is defined as a variable whose values are linguistic terms rather than numbers, while the fuzzy conditional statements are expressions of the form IF *A* THEN *B*.

Since then, the literature on fuzzy logic theory and control has been growing rapidly, which illustrated its useful implication in many industrial applications. Early implementation of fuzzy logic control (FLC) was described by Mamdani (1974), which demonstrated the feasibility of FLC for a simple dynamical plant featured in a model steam engine. In 1977, Mamdani and Pappis described the implementation of FLC for a traffic junction controller. Takagi and Sugeno (1985) have developed a mathematical tool to build a fuzzy model of a system where fuzzy implications and reasoning are used. Furthermore, two practical applications for using FLC have been also discussed in the same publication: water cleaning process and a converter in a steel-making process.

These early developments of fuzzy logic have been served to trigger the very rapid and widespread propagation of interest in using FLC in several applications. The key success of FLC lies in its ease of implementation, as it usually needs no mathematical model of the controlled system. However, a graphical representation of the system behaviour is essential in the FLC design. It is mainly based on a

logical model, which simulates the thinking process of a human operator while controlling the system manually. In a conventional controller such as the classical PID, on the other hand, a precise mathematical model has to be obtained to design the gains of the controller. Furthermore, the classical PID controller with fixed gains cannot generate an optimal response in a plant parameter varying system. Hence, performance will degrade solely because of system non-linearity and parameters variation. In other word, the controller gains have to be frequently readjusted to cope with the changes in the system parameters.

Indeed, the superiority of FLC systems over the traditional PID controller has been discussed in literature, whereby they work fairly well with varying controlled system parameters (Liaw and Wang 1991; Fodor et al. 1996; Lai et al. 1996; Da Silva and Acarnley 1997; Heber et al. 1997; Vas et al. 1997). Therefore, they don not require redesign whenever sudden changes occur in these parameters. It is for this reason that many researchers and manufacturers now implement FLC in high performance applications where only PID controllers were previously available.

For further readings on the development of FLC techniques, an overall survey of the FLC methodology is well overviewed by Lee (1990a; 1990b), while a survey of recent industrial applications of FLC is presented by Schwartz et al. (1994) and Bonissone et al. (1995). The theoretical development and application of FLC are thoroughly reviewed by Dubois and Prade (1980), Aminzadeh and Jamshidi (1994), Kruse et al. (1994), and Mandel (1995). The combinations of FLC with other control algorithms such as the conventional PID, neural network and the variable structure control are presented by many authors (Tzafestas and Papanikolopoulos 1990; Ishigame et al. 1993; Suyitno et al. 1993; Jang and Sun 1995; Baghli et al. 1997).

In this chapter, a review of recent applications of FLC in ac speed drives is presented. This is followed by a brief summary of some of the relevant concepts in fuzzy set theory and fuzzy logic. The main ideas underlying the FLC are reviewed. The concept of solving any control problem by a set of linguistic rules is highlighted. The step-by-step procedures to build a FLC controller to regulate the speed of the induction motor are presented. The control behaviour of FLC is simulated in MATLAB and SIMULINK, in which the fuzzy rules are obtained according to the control experience of the induction motor drive presented by Liaw and Wang (1991).

4.2 Recent applications of FLC in induction motor drives:

With no doubts, the possibility of introducing FLC in high performance ac speed drives is valid, and as a result, it has recently attracted a significant attention in the research community worldwide. Early use of FLC in the induction motor drive applications has been presented by Liaw and Wang (1991). Through the simulation and experimental results, they proved that good dynamics and speed response can be achieved by employing FLC in the speed control feedback loop. Moreover, it has been also shown that due to the highly adaptive capability possessed by FLC, the performance of the drive system is quite robust and insensitive to parameter and operating condition changes.

Since then, the application of FLC to improve the operating characteristics of the induction motor drives has been reported by many authors. Cheng and Yeh (1993) presented a novel FLC to be used in a fully digital speed control of an ac servo system. They proposed a simple speed control structure, where all the control algorithms featured in FLC are computed by a 16-bit single-chip microprocessor, whereby the number of circuit components is reduced and thus more cost-effective.

Kung and Liaw (1994) proposed rather interesting control algorithms incorporating FLC to improve the performance of the induction motor drive with the use of a linear model following controller. Under this scheme of control, the motor speed can be regulated to follow the output speed produced by a motor reference model, in which it is selected to generate the desired dynamic tracking response. As a result, instead of using the error between the actual motor speed and the desired speed, the error between the actual speed and the output of the reference model is used to drive the FLC. The simulated and experimental results have indicated a good speed tracking response.

Fuzzy logic control has been also used for the implementation of direct self control (DSC) of induction motor to improve its slow response as reviewed by Mir et al. (1994). In DSC, the torque and stator flux are regulated to their command values by selecting the switching states, which are based on the error in electric torque, error in stator flux and the position of the stator flux vector. It followed from the experimental results that the dynamical performance of the induction motor with FLC showed a considerable improvement over the conventional control.

The improvement of variable speed drive system efficiency using FLC is presented by Sousa et al. (1995), where an on-line efficiency optimisation control is proposed for an indirect vector controlled induction motor drive. In theory, the best transient response is obtained when the drive system is operating at rated conditions, in which the rated flux is reached. However, at light loads, rated flux operation causes excessive core loss, thus impairing the efficiency of the drive. The authors argued that this problem is solved by using FLC that adaptively decrements the excitation current on the basis of the measured input power. Thus for a given load torque and speed, the drive settles down to the minimum input power and hence operates at maximum efficiency.

Application of FLC in the phase-locked loop (PLL) speed control of induction motor drive is presented by Lai et al. (1996). The proposed system combines the good speed regulation of the PLL techniques (Moffat et al. 1979; Mittal and Ahmed 1983) and the advantages of the FLC to obtain a robust, fast and precise control of the motor speed. In this system, the induction motor is driven by the FLC towards the PLL locking range whenever the error between the reference speed and the actual speed of the motor is larger than a pre-set error value. Once the drive system is operating in the locking range, the FLC is replaced by the PLL control, in which the actual motor speed is synchronised with the reference speed. It followed from the experimental results that the performance of the induction motor has been virtually improved by combining the rapidity of the FLC during the transient period with the accuracy of the PLL during the steady state.

Wang and Liaw (1997) proposed two fuzzy logic controllers to be used in field-oriented induction motor speed drive. It is well known that the static and dynamic operating performances of the indirect field-oriented speed drives are greatly affected by its decoupling characteristic, which depends on accurate motor parameters identification. However, heating and saturation effects will cause the parameters to change, and as a result the decoupling between flux and torque is lost and both steady state and transient response are degraded. Thus the main task of the first FLC is to on-line tune the motor slip gain for achieving ideal decoupling control. On the other hand, the second FLC is employed to perform flux level control so that maximum operating efficiency is obtained. Inspection of experimental results showed some effective improvements in operating performances and dynamic speed responses of the induction motor drive.

Vas et al. (1997) discussed the application of fuzzy control system in a DSP-based fully digital high dynamic performance induction motor drive. The high performance drive considered is a vector drive system using rotor flux oriented control and space vector PWM modulator. The input signals to the modulator are generated by a control system, which contains four interacting fuzzy controllers: the speed, the flux, the torque producing current and flux producing current controllers. The paper discussed the design of the four controllers, and both simulation and experimental results are given. The results show that the fuzzy controllers provide high dynamic performance when compared to those for a drive containing classical PI controllers.

A performance comparison between fuzzy and conventional PI speed control within vector control for induction motor is presented by Bahli et al. (1997). Experimental results on position control and speed control under load are demonstrated. The obtained results illustrated the superiority of the fuzzy controller over the conventional IP anti windup controller in fast tracking position and speed commands.

Very recently, Heber et al. (1997) presented a single and relatively simple FLC that is capable to enhance the performance of the indirect field oriented induction motor drive in terms of speed tracking, external disturbance rejection and parameter variations. With the aid of simulated and experimental results, performance comparison between the conventional PID and the proposed FLC has been made. Inspection of these results illustrated that a properly designed FLC can outperform the classical PID both when the motor is properly field oriented and when it becomes detuned.

Another paper has been published by Zhen and Xu (1998), in which an on-line fuzzy tuning scheme for indirect field oriented control has been presented. In the proposed system, one FLC is used to control the speed of the induction motor, and another two fuzzy compensators are combined to correct the detuning of field orientation. It followed from the experimental results that the proposed control scheme reduced the detuning effects and effectively improved the performance of the induction motor drive.

4.3 Fuzzy sets and fuzzy logic, a summary of relevant concepts:

Fuzzy set and fuzzy logic are powerful mathematical tools used to model uncertain system behaviour and facilitate the control design in the absence of complete and precise information about the controlled system. It will be convenient to summarise the relevant properties of the fuzzy set theory and the basic concepts of fuzzy logic that are aimed to aid the development of FLC in later sections. In what follows, all letters and terms in italic are referred to notation and properties of fuzzy sets, respectively.

A. Definition of fuzzy sets and terminology:

A *fuzzy set A* of a universe of discourse U is characterised by a membership function $\mu_A(x)$: $U \rightarrow [0,1]$, where $\mu_A(x)$ specifies the grade or degree to which any element x in U belongs to the fuzzy set A. The general definition of fuzzy set A can be obtained as:

$$A = \left\{ \left(x, \mu_A(x)\right) \mid x \in U,\ \mu_A(x) \in [0,1] \right\} \tag{4.1}$$

Definition (4.1), associates with each element x in U a real number $\mu_A(x)$ in the interval between [0,1], which represents the degree of membership of x in A. Larger values of $\mu_A(x)$ indicate higher degrees of membership and vice versa. Figure 4-1 shows a characteristic example of a membership function that maps elements of U to a membership range which is usually in the interval between [0,1].

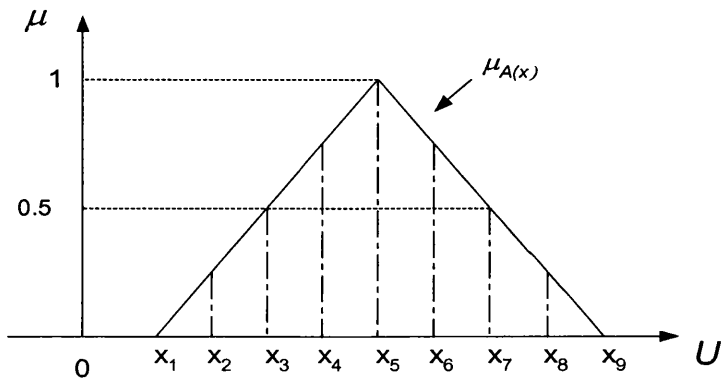


Figure 4-1: Membership function of fuzzy set A.

It follows from Figure 4-1 that the fuzzy set A can be represented concisely by the summation

$$A = \mu_1 / x_1 + \ldots\ldots + \mu_n / x_n \tag{4.2}$$

or

$$A = \sum_{i=1}^{n} \mu_i / x_i \tag{4.3}$$

in which μ_i, $i = 1,\ldots,n$, is the degree of membership of x_i in fuzzy set A. It should be noted that the + sign in (4.2) and (4.3) refers to the set union rather than the arithmetic summation, while the / sign is used to connect each element with its membership value and has no connection to the arithmetic division.

The *support* of fuzzy set A is the set of all elements x (x_1, x_2, ...,x_n) in the universe of discourse U, at which $\mu_A(x) > 0$. The *crossover point* in fuzzy set A is an element (or elements) x in U whose degree of membership in A is 0.5 (such as x_3 and x_7 in Figure 4-1). A fuzzy set A whose support is just a single element x_1 and $\mu_A(x_1) \in [0,1]$ is called *fuzzy singleton* (Lee 1990a).

B. Basic theoretic operations on fuzzy sets:

For simplicity of illustration, two fuzzy sets A and B in the universe of discourse U with membership functions μ_A and μ_B, respectively, are considered in which

$$A = \left\{ \left(x, \mu_A(x)\right) \right\}, \ \mu_A(x) \in [0,1] \quad , x \in U \tag{4.4}$$

$$B = \left\{ \left(x, \mu_B(x)\right) \right\}, \ \mu_B(x) \in [0,1] \quad , x \in U \tag{4.5}$$

It should be noted that the set operations with A and B are performed based on operations on their membership functions as in the following,

Definition 1: Equality: The fuzzy sets A and B are *equal* denoted by $A = B$ if and only if for every element x in U,

$$\mu_A(x) = \mu_B(x) \qquad , x \in U \tag{4.6}$$

Definition 2: Complementation: The fuzzy sets A and $\overline{A}$ are complementary for all element x in U if

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \qquad , x \in U \tag{4.7}$$

Definition 3: Intersection: The membership function $\mu_{A \cap B}$ of the intersection operation between the fuzzy sets denoted as $A \cap B$ is defined for every element x in U by

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \qquad , x \in U \tag{4.8}$$

Definition 4: Union: The membership function $\mu_{A \cup B}$ of the union operation between the fuzzy sets denoted $A \cup B$ is defined for every element x in U by

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \qquad , x \in U \tag{4.9}$$

Definition 5: Algebraic Product : The membership function $\mu_{A \cdot B}$ of the algebraic product operation between the fuzzy sets denoted $A \cdot B$ is defined for every element x in U by

$$\mu_{A \cdot B}(x) = \mu_A(x) * \mu_B(x) \qquad , x \in U \tag{4.10}$$

Definition 6: Cartesian Product : If A and B are fuzzy sets in the universe of discourse U, the Cartesian product (or cross product) denoted $A \times B$ is the set of ordered pairs given as

$$A \times B = \{(a,b) \mid a \in A,\ b \in B\} \tag{4.11}$$

C. Fuzzy relations and compositional operators:

A *fuzzy relation R* from a fuzzy set A to a fuzzy set B is a fuzzy subset of the Cartesian product $A \times B$, where A and B are subsets of the universes U_1 and U_2, respectively. This leads to the definition of the *fuzzy relation R* as

$$R = \left\{ \left((x,y), \mu_R(x,y)\right) \mid (x,y) \in A \times B,\ \mu_R(x,y) \in [0,1] \right\} \tag{4.12}$$

where the membership function $\mu_R(x,y)$ gives the degree of membership of the ordered pair (x,y) in the fuzzy relation R. In other words, this degree of membership indicates the degree to which x is in relation with y.

If R is a relation from fuzzy set A to fuzzy set B and S is a relation from fuzzy set B to fuzzy set C, then the *composition* of R and S results in a fuzzy relation denoted by $R \circ S$ and defined by

$$R \circ S = \left\{ \left((x,z),\ \max_y \left(\mu_R(x,y) * \mu_S(y,z) \right) \right) \right\} \tag{4.13}$$

where * denotes a compositional operators such as *intersection* (*min*), *product*, *bounded product* and *drastic product*. Suitable selection of any of these operators depends highly on the designer, where it is chosen to fit a specific application. In many FLC applications, the *max-min* and the *max-product* compositional operators are the most frequently used due to their ease of implementation (Lee 1990a). More detailed discussions on fuzzy relation and compositional rules of inference are presented by Dubois and Prade (1980) and Kruse et al. (1994)

4.4 The principles of fuzzy logic control:

A precise definition that describes the behaviour of FLC does not exist. However the basic ideas underlying its mathematical structure can be explained. Its dynamic behaviour is characterised by a set

of *linguistic variables* and *fuzzy rules*, which are usually called fuzzy conditional statements. In other words, FLC is a mathematical set theory, which allows systems to be controlled according to the human intuition and experience using fuzzy rules in the form of *IF-THEN* conditional statements. Figure 4-2 shows the basic configuration of a typical FLC that comprises three elements: a *Fuzzifier*, a *Fuzzy Knowledge Base with Inference Engine* and a *De-fuzzifier*.

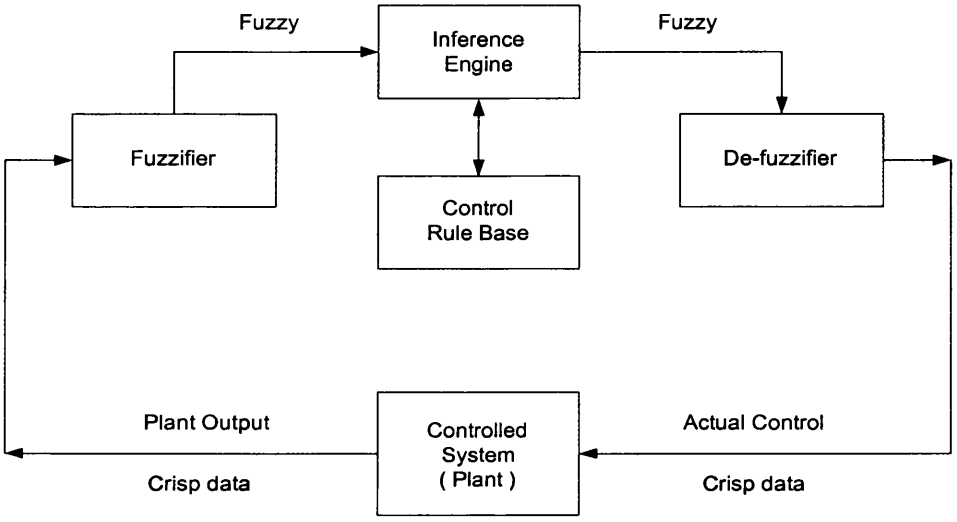


Figure 4-2: Block diagram of a typical FLC.

A. Fuzzification process:

The basic function of the fuzzification stage is to convert the crisp input variable into a fuzzy singleton within a certain universe of discourse. In FLC applications, the acquired input variables to the controller are usually crisp in nature. Therefore, fuzzification is necessary at first since the data manipulation in an FLC is based on fuzzy set theory. This can be accomplished by mapping the crisp values of the input variable through all the fuzzy sets over the corresponding universe of discourse. The mapped data are further converted into a fuzzy singleton which basically gives the degree of membership for each crisp value in relation with all fuzzy sets within the corresponding universe of discourse. The fuzzification process can be expressed by

$$x = fuzzifier\left(x_0\right) \tag{4.14}$$

where x_0 and x represent the crisp input value from a plant and the pre-defined fuzzy set for the input variable, respectively.

B. Fuzzy knowledge with inference engine:

In FLC applications, the fuzzy knowledge base consists of two components known as a *data base* and a *fuzzy control rule base*. The data base provides essential information which is used to define the membership functions of the fuzzy sets for each system variable (inputs and outputs) and to characterise the data manipulation in an FLC. On the other hand, the fuzzy control rule base contains a collection of fuzzy linguistic rules intended to achieve the control objectives. These rules are basically developed based on the expert knowledge of the controlled plant. Their structure is usually in the form of IF-THEN conditional statements. It should be noted that the proper selection of the control variables and the shape of the membership function is essential to develop quite useful fuzzy rules and hence optimise the overall operational performance of the FLC system. Once the knowledge base has been constructed, modelling the human decision making within the framework of FLC is completed by employing an inference engine. Here the consequents from the designed fuzzy rules are combined (typically union via MAX) to provide the inferring fuzzy control action.

C. *Defuzzification process:*

This process is needed to produce a non-fuzzy (crisp) control action that best represents the distribution of the inferred fuzzy control action. This can be expressed by

$$y_0 = defuzzifier\left(y\right) \tag{4.15}$$

where y and y_0 are fuzzy and crisp control actions, respectively. There exist several de-fuzzification methods. Some of the earliest FLCs used the "*mean of maxima*", meaning that the crisp output is taken as the value at which the membership function of the inferred fuzzy set reached its maximum, but if there are many maxima the mean of these is taken. In the present fuzzy controllers, the *"centroid"* or *"centre of gravity"* method is the most commonly used. It can be computed by defining the contour or aggregate of the inferred control action which is used to solve the following discrete formula,

$$y_0 = \frac{\sum_{k=1}^{N} y_k * \mu_{agg}(y_k)}{\sum_{k=1}^{N} \mu_{agg}(y_k)} \tag{4.16}$$

where y_k represent the subdivided discrete samples into N equal subintervals of the output universe of discourse, while $\mu_{agg}(y_k)$ contain the corresponding aggregated degree of membership for each sample.

D. Fuzzy linguistic variables:

Linguistic variables are defined as variables whose values are words or sentences in natural or artificial languages. They play an essential role in providing a systematic mean for an approximate characterisation of complex or ill-defined systems. For example, by employing the concept of fuzzy set theory, the motor *speed error* can be described approximately. The *speed error* is a linguistic variable comprising some fuzzy sets such as *NB (Negative Big), NS (Negative Small), ZO (Near Zero), PS (Positive Small),* and *PB (Positive Big).* In other words, the attribute *speed error* is a *fuzzy variable*, whose values are *linguistic labels* of fuzzy sets. Each set is defined by an appropriate membership function as shown in Figure 4-3, where a normalised universe [+1,-1] is being used. Membership functions having the forms of triangle-shaped and trapezoid-shaped are used in here due to their ease of implementation compared to other existing functions. It is important to note that the characterisation of a value of the fuzzy variable *speed error* by a fuzzy label like *PS* is less precise than its exact numerical value when it is taken as revolution per minute (r.p.m).

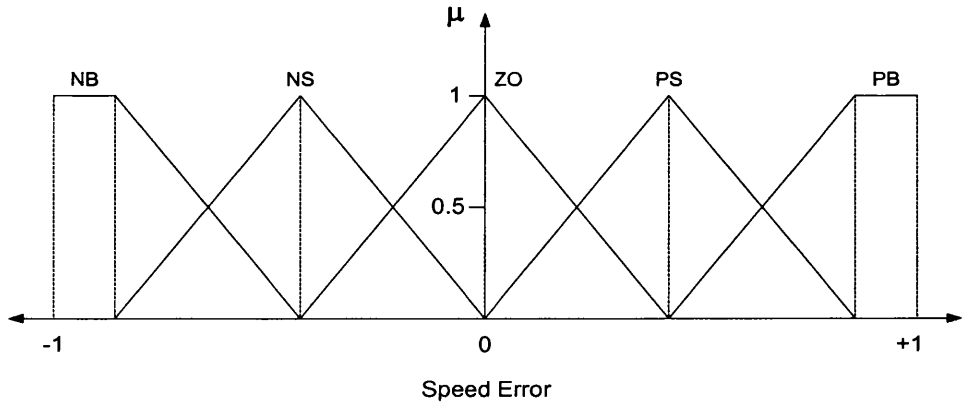


Figure 4-3: Membership functions for the linguistic variable speed error.

It should be also noted that the number of fuzzy sets and the correct selection of the membership function have a substantial effect on the performance of the FLC. Moreover, the proper choice of the plant state variables and control variables is essential to characterise the operation of the fuzzy system. And indeed, the essence of the FLC performance depends strongly on the derivation of useful fuzzy control rules.

E. Fuzzy control rules:

In an FLC, the performance of the fuzzy system is characterised by a set of linguistic rules that are developed according to the expert knowledge of the controlled system. In most engineering control applications, these rules are simply expressed in the form of "IF-THEN" conditional statements, e.g.:

$$Rule_i: \text{ IF } x \text{ is } A_i \text{ AND } y \text{ is } B_i \text{ THEN } z \text{ is } C_i$$

where x, y and z are linguistic variables representing the plant state variables and the output control variables. A_i, B_i and C_i are fuzzy labels of the linguistic variables x, y, and z in the universes of discourse U, V, and W, respectively. In this rule, the IF part is known as the *antecedent*, while the THEN part is known as the rule *consequent*. Hence, the fuzzy rule is a conditional statement in which the antecedent is a condition and the consequent is a control action for the system under control.

The number of fuzzy rules is determined by computing the product of the number of fuzzy sets in each input linguistic variables, x and y. These rules are then designed to produce as a conclusion l different outputs, where l is the number of fuzzy sets in the output linguistic variable, z. For example, in the case of two-input single-output fuzzy system, where both input variables have the same number of fuzzy sets 5, there should be 5*5 = 25 fuzzy rules.

In many cases, large number of rules can be reduced using several methods (Lee 1990b). One method is achieved by the reduction of the antecedent (linguistic) variables or their fuzzy sets. The other method is to keep track of the rules being computed during real-time or off-line inferencing of the FLC and try to eliminate those rules, which are either not used at all or used very little.

4.5 Computer model of fuzzy logic control for ac speed drives:

The control objectives for a high performance speed drive system are: 1) quick tracking performance of command speed changes without overshoot, and 2) the maximum speed drop and the restore time when load is applied must be kept as small as possible. These objectives can be achieved by using FLC, since the dynamic drive system model is not necessary for its design. Moreover, the control performance is insensitive to parameters and operating condition changes. In what follows, a simple example of fuzzy logic controller for variable speed drive system is described. With the computer model developed in MATLAB for the FLC design, simulations of the control concepts were first carried out for testing purposes. This model has been constructed in M files, whereby it enables an easy design and redesign of the FLC system to meet the control requirements.

A. Initial design of FLC and derivation of fuzzy linguistic rules:

In the proposed fuzzy controller, the process variable to be observed and the control variable of the motor drive are chosen to be the rotor speed ω_r and the torque current i_{sq}^*, respectively. The actual inputs to the fuzzy system are the speed error (the error between the rotor speed and the command speed) and its rate of change as defined in the following equations

$$e_0(k) = \omega_r^*(k) - \omega_r(k) \tag{4.17}$$

$$\Delta e_0(k) = e_0(k) - e_0(k-1) \tag{4.18}$$

where $\omega_r^*(k)$ is the speed command in kth sampling interval, $\omega_r(k)$ is the rotor speed response in the kth sampling interval, $e_o(k)$ is the speed error in the kth sampling interval, and $\Delta e_o(k)$ is the speed error change in the kth sampling interval. On the other hand, the output of the FLC is the current command change Δi_{sq}^*, which is integrated at regular kth sampling intervals and yields the following current command,

$$i_{sq}^*(k) = i_{sq}^*(k-1) + G_0 \Delta i_{sq}^*(k) \tag{4.19}$$

where the integral gain G_0 can be varied to tune the output of the fuzzy controller for a desired performance. To determine the FLC output for a given error and its rate of change, the linguistic control rules should be defined. These rules are developed based on the expert experience and control engineering knowledge of the drive system. In order to obtain a good dynamic performance, the following concepts should be considered:

1) Depending on the magnitudes and signs of the speed error and its rate of change, the FLC increments or decrements the torque current command such that the rotor speed reaches the speed command.

2) The FLC maintains the output value when the error and its rate of change are zero. If any disturbances occur such as sudden external load and parameters perturbation, the FLC should quickly change the current command to keep the rotor speed at the speed command.

Based on theses concepts, the linguistic control rules are developed and listed in Table 4.1, as proposed by Liaw and Wang (1991). It is seen that each linguistic variable (error, rate of change of error and controller output) is represented in fuzzy set notations using the linguistic labels NB, NM, NS, ZO, PS, PM and PB. It follows from this table (also known as the decision table) that there are 49 combinations of speed error $e_o(k)$ and its rate of change $\Delta e_o(k)$, where each one of them corresponds to a particular rule. Thus there are 49 conditional rules on which the FLC output is decided. For example, the element in the first row and fifth column reads the following rule

Rule 5: **if** ($e_o(k)$ is NB ***and*** $\Delta e_o(k)$ is PS) **Then** (the controller output $\Delta i_{sq}^*(k)$ is NM)

B. Design of membership functions:

Following the construction of the decision table, the membership function for each fuzzy set must be defined. This can be achieved by assigning the grade of membership to each element of the corresponding universe of discourse. There exist several types of membership functions such as the bell-shaped, the trapezoidal-shaped and the triangular-shaped. The choice of membership function is mainly dependent on particular applications. In this simulation, the trapezoidal-shaped functions are used to characterise the membership functions.

ERROR $e_o(k)$	RATE OF CHANGE OF SPEED ERROR $\Delta e_o(k)$						
	NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
NB	ZO	NS	NS	NM	NM	NB	NB
NM	PS	ZO	NS	NS	NM	NM	NB
NS	PS	PS	ZO	NS	NS	NM	NM
ZO	PM	PS	PS	ZO	NS	NS	NM
PS	PM	PM	PS	PS	ZO	NS	NS
PM	PB	PM	PM	PS	PS	ZO	NS
PB	PB	PB	PM	PM	PS	PS	ZO

Table 4.1: The decision table

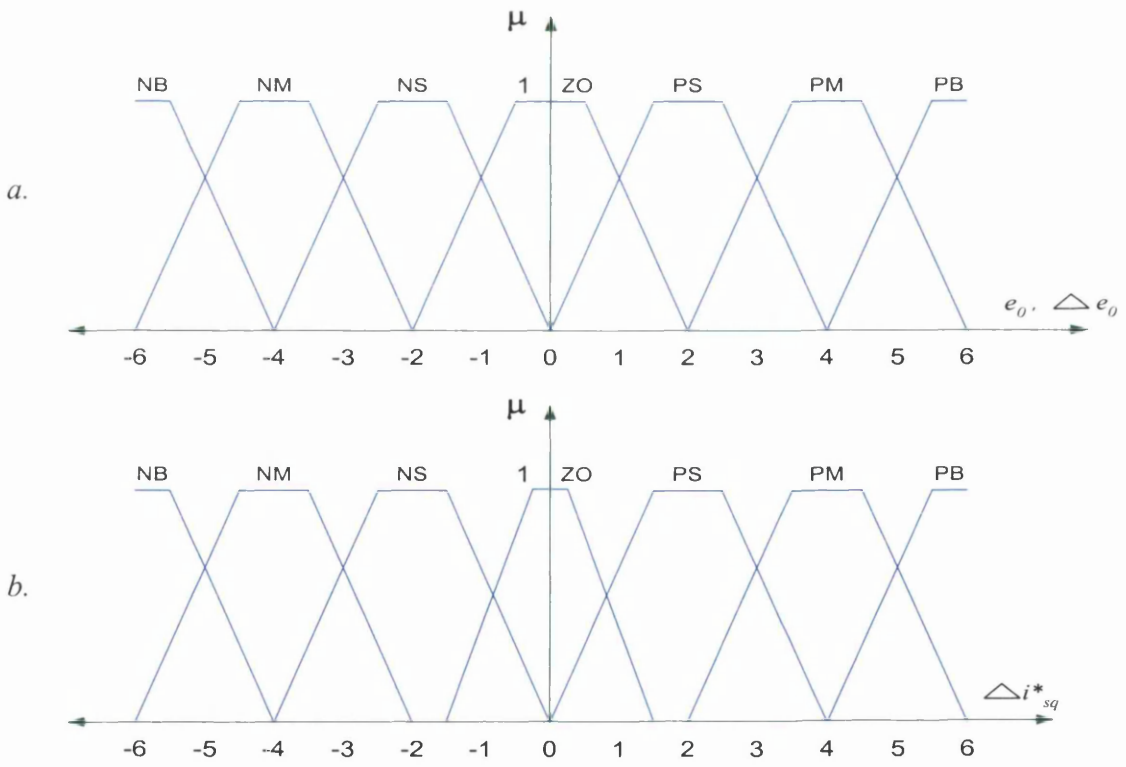


Figure 4-4: Membership functions representing the FLC variables in linguistic labels: (a) input membership functions; (b) output membership functions

Figure 4-4 shows the simulated fuzzy sets for the input (e_o and Δe_o) and output (Δi_{sq}^*) linguistic variables. For the purpose of digital computer processing, it is seen that the universe of discourse has been discretised into thirteen equal segments (-6, -5,..., 5, 6). In theory, discretisation of the universe is normally referred to as *quantisation* and the divided segments are known as *quantisation levels*. The choice of these levels has a tremendous influence on the FLC resolution. In other words, the number of quantisation levels should be large enough so that the FLC is more sensitive to the observed variables and hence provides fine control output.

The quantised levels of e_o and Δe_o are shown in Table 4.2, in which the error and change of error are chosen from –300 to 300 rpm in a step of 50 and from –90 to 90 rad/sec^2 in a step of 15, respectively. The control output is also discretised into thirteen levels (also shown in Table 4-2).

Quantised level	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
e_o (r/min)	-300	-250	-200	-150	-100	-50	0	50	100	150	200	250	300
Δe_o (rad/sec^2)	-90	-75	-60	-45	-30	-15	0	15	30	45	60	75	90
Δi_{sq}^*	-45	-37	-30	-23	-15	-5	0	5	15	23	30	37	45

Table 4.2: The quantised levels of the error and its rate of change.

C. *Fuzzy relations and inference methods:*

The relationship between the fuzzy set characterising the input variables and the fuzzy set characterising the controller output can be simplified using the fuzzy set theory (Zadeh 1965). This relationship is easily constructed by converting the decision table to a *fuzzy relation table* as laid out in Table 4.3. Basically, the construction of such table is mainly based on the designed linguistic rules. In other words, for a particular fuzzy rule a membership degree of 1 is assigned to the contributed output fuzzy set, while a degree of 0.5 is assigned to the adjacent sets and 0 degree to the rest.

x_i	CONTROLLER INPUTS $(e_o, \Delta e_o)$	CONTROLLER OUTPUT (MEMBERSHIP GRADES) $\mu(x_i, NB)$	$\mu(x_i, NM)$	$\mu(x_i, NS)$	$\mu(x_i, ZO)$	$\mu(x_i, PS)$	$\mu(x_i, PM)$	$\mu(x_i, PB)$
x_1	(NB, NB)	0	0	0.5	1	0.5	0	0
x_2	(NB, NM)	0	0.5	1	0.5	0	0	0
x_3	(NB, NS)	0	0.5	1	0.5	0	0	0
x_4	(NB, ZO)	0.5	1	0.5	0	0	0	0
x_5	(NB, PS)	0.5	1	0.5	0	0	0	0
x_6	(NB,PM)	1	0.5	0	0	0	0	0
x_7	(NB, PB)	1	0.5	0	0	0	0	0
x_8	(NM, NB)	0	0	0	0.5	1	0.5	0
x_9	(NM, NM)	0	0	0.5	1	0.5	0	0
x_{10}	(NM, NS)	0	0.5	1	0.5	0	0	0
x_{11}	(NM, ZO)	0	0.5	1	0.5	0	0	0
x_{12}	(NM, PS)	0.5	1	0.5	0	0	0	0
x_{13}	(NM,PM)	0.5	1	0.5	0	0	0	0
x_{14}	(NM, PB)	1	0.5	0	0	0	0	0
x_{15}	(NS, NB)	0	0	0	0.5	1	0.5	0
x_{16}	(NS, NM)	0	0	0	0.5	1	0.5	0
x_{17}	(NS, NS)	0	0	0.5	1	0.5	0	0
x_{18}	(NS, ZO)	0	0.5	1	0.5	0	0	0
x_{19}	(NS, PS)	0	0.5	1	0.5	0	0	0
x_{20}	(NS,PM)	0.5	1	0.5	0	0	0	0
x_{21}	(NS, PB)	0.5	1	0.5	0	0	0	0
x_{22}	(ZO, NB)	0	0	0	0	0.5	1	0.5
x_{23}	(ZO, NM)	0	0	0	0.5	1	0.5	0
x_{24}	(ZO, NS)	0	0	0	0.5	1	0.5	0
x_{25}	(ZO, ZO)	0	0	0.5	1	0.5	0	0
x_{26}	(ZO, PS)	0	0.5	1	0.5	0	0	0
x_{27}	(ZO,PM)	0	0.5	1	0.5	0	0	0
x_{28}	(ZO, PB)	0.5	1	0.5	0	0	0	0
x_{29}	(PS, NB)	0	0	0	0	0.5	1	0.5
x_{30}	(PS, NM)	0	0	0	0	0.5	1	0.5
x_{31}	(PS, NS)	0	0	0	0.5	1	0.5	0
x_{32}	(PS, ZO)	0	0	0	0.5	1	0.5	0
x_{33}	(PS, PS)	0	0	0.5	1	0.5	0	0
x_{34}	(PS,PM)	0	0.5	1	0.5	0	0	0
x_{35}	(PS, PB)	0	0.5	1	0.5	0	0	0
x_{36}	(PM, NB)	0	0	0	0	0	0.5	1
x_{37}	(PM, NM)	0	0	0	0	0.5	1	0.5
x_{38}	(PM, NS)	0	0	0	0	0.5	1	0.5
x_{39}	(PM, ZO)	0	0	0	0.5	1	0.5	0
x_{40}	(PM, PS)	0	0	0	0.5	1	0.5	0
x_{41}	(PM,PM)	0	0	0.5	1	0.5	0	0
x_{42}	(PM, PB)	0	0.5	1	0.5	0	0	0
x_{43}	(PB, NB)	0	0	0	0	0	0.5	1
x_{44}	(PB, NM)	0	0	0	0	0	0.5	1
x_{45}	(PB, NS)	0	0	0	0	0.5	1	0.5
x_{46}	(PB, ZO)	0	0	0	0	0.5	1	0.5
x_{47}	(PB, PS)	0	0	0	0.5	1	0.5	0
x_{48}	(PB,PM)	0	0	0	0.5	1	0.5	0
x_{49}	(PB, PB)	0	0	0.5	1	0.5	0	0

Table 4.3: The fuzzy relation table.

For example, the conditional part of Rule 5 reads as follows:

$$\text{Rule 5:} \quad \textbf{if} \ (\ e_o(k) \text{ is NB } \textbf{\textit{and}} \ \Delta e_o(k) \text{ is PS})$$

then the controller output can be described by the following fuzzy sets

$$[(\text{NB}, 0.5), (\text{NM}, 1), (\text{NS}, 0.5), (\text{ZO}, 0), (\text{PS}, 0), (\text{PM}, 0), (\text{PB}, 0)]$$

The same procedure is then repeated for the remaining rules, and hence the relationship between the controller inputs and output variables is defined. Having done that, the final fuzzy control action is then determined by adopting the following steps:

Step 1: The degree of membership for the conditional part in each fuzzy rule is determined. The *firing strength* of each contributed rule to the fuzzy control action is thus obtained. In most of the existing FLC systems, the conditional part is usually combined by an '***and***' operator, which is equivalent to the '**min**' operator. As an illustration, using the law of intersection between two fuzzy sets as defined in Equation (4.8), the firing strength of rule 5 can be determined as follows:

$$\begin{aligned} \mu(x_5) &= \mu(e_o(k) \text{ is NB } \textbf{\textit{and}} \ \Delta e_o(k) \text{ is PS}) \\ &= \textbf{min}\ (\mu(e_o(k) \text{ is NB}),\ \mu(\Delta e_o(k) \text{ is PS})) \end{aligned} \tag{4.20}$$

Step 2: Having defined the degree of membership for the conditional part using equation (4.20) and the fuzzy relation table, the membership values for the control output characterised by the linguistic terms 'NB, NM, NS, ZO, PS, PM, PB' can be determined by two methods. The first uses the law of intersection given in Equation (4.8), which is widely used in FLC applications. For example, the degree of membership for the linguistic term NM can be computed as follows

$$\mu_5(NM) = \textbf{min}\ (\mu(x_5, \text{NM}), \mu(x_5)) \tag{4.21}$$

The second employs the algebraic product given in Equation (4.10), thus the membership grade for 'NM' can be defined as follows,

$$\mu_5 (NM) = (\mu (x_5, \text{NM}) * \mu (x_5)) \tag{4.22}$$

Step 3: The method defined either in (4.21) or (4.22) is then repeated for all the 49 rules, thus the final degree of membership for the fuzzy control output can be evaluated using the composition rule of the fuzzy set theory given in equation (4.13). For example, the inferred fuzzy set characterised by 'NM' can be computed as follows

$$\mu (NM) = \textbf{max} (\textbf{min} (\mu (x_i, \text{NM}), \mu (x_i))), \quad i = 1, 2, \ldots, 49 \tag{4.23}$$

or

$$\mu (NM) = \textbf{max} ((\mu (x_i, \text{NM}) * \mu (x_i))) \quad , \quad i = 1, 2, \ldots, 49 \tag{4.24}$$

This procedure is then repeated for the fuzzy control output characterised by the other linguistic terms 'NB, NS, ZO, PS, PM, PB', which indicates the contribution of each output fuzzy set to the inferred fuzzy control action. Since the fuzzy logic controller output can only be a crisp (numerical) value, the resulted fuzzy control action expressed in linguistic terms has to be defuzzified. The centre of gravity method as given in Equation (4.16) is used throughout to perform the defuzzification process.

D. Simulation results:

To demonstrate the effectiveness of the designed fuzzy control action, two numerical values of error and change of error are selected to be the controller input signals, $e_o = 70$ r/min and $\Delta e_o = -25$ rad/sec^2. Fuzzification is then performed on these input signals by mapping them through all the fuzzy sets over the corresponding universe of discourse using the reference fuzzy sets defined in Figure 4-4. As a result, the controller inputs can be described by the following fuzzy singletons,

e_o = { (NB, 0.0), (NM, 0.0), (NS, 0.0), (ZO, 0.4615), (PS, 1.0), (PM, 0.0), (PB, 0.0) }

Δe_o = { (NB, 0.0), (NM, 0.0), (NS, 1.0), (ZO, 0.250), (PS, 0.0), (PM, 0.0), (PB, 0.0) }

where the numbers define the degree of membership of each contributed input fuzzy set in relation with the corresponding crisp input signal. The membership grades of the conditional parts for all the linguistic rules are defined from Equation (4.20) as follows,

$$\mu(x_1) = \mu(e_o(k) \text{ is NB } \textbf{and } \Delta e_o(k) \text{ is NB})$$
$$= \textbf{min}\,(\mu(e_o(k) \text{ is NB}),\ \mu(\Delta e_o(k) \text{ is NB}))$$
$$= \textbf{min}\,(0, 0) = 0$$

.

.

$$\mu(x_{49}) = \mu(e_o(k) \text{ is PB } \textbf{and } \Delta e_o(k) \text{ is PB})$$
$$= \textbf{min}\,(\mu(e_o(k) \text{ is PB}),\ \mu(\Delta e_o(k) \text{ is PB}))$$
$$= \textbf{min}\,(0, 0) = 0$$

Next, using the intersection rule of fuzzy sets and the fuzzy relation table, the degree of membership for the output fuzzy sets characterised by NM can be computed for all the fuzzy rules as follows,

$$\mu_1(NM) = \textbf{min}\,(\mu(x_1, \text{NM}), \mu(x_1))$$
$$= \textbf{min}\,(0, 0) = 0$$

.

.

$$\mu_{49}(NM) = \textbf{min}\,(\mu(x_{49}, \text{NM}), \mu(x_{49}))$$
$$= \textbf{min}\,(0, 0) = 0$$

The final degree of membership for the fuzzy logic control output NM is then determined using the composition rule from Equation (4.23) as follows,

$$\mu(NM) = \textbf{max}\,(\textbf{min}\,(\mu(x_i, \text{NM}), \mu(x_i))), \qquad i = 1, 2, \ldots, 49$$
$$= 0$$

Finally, the degree of membership for the other six fuzzy control output characterised by 'NB, NS, ZO, PS, PM, PB' can be computed in exactly the same way. Thus the inferred membership grades are as follows,

$$\mu(NB) = 0; \mu(NS) = 0.25; \mu(ZO) = 0.5; \mu(PS) = 1.0; \mu(PM) = 0.5; \mu(PB) = 0$$

Using the centre of gravity defuzzification method, the inferred fuzzy control action, as depicted in Figure 4-5, is converted to a numerical value, which represents the change in the torque current command (Δi_{sq}^{*}). The defuzzified output is shown in Figure 4-6.

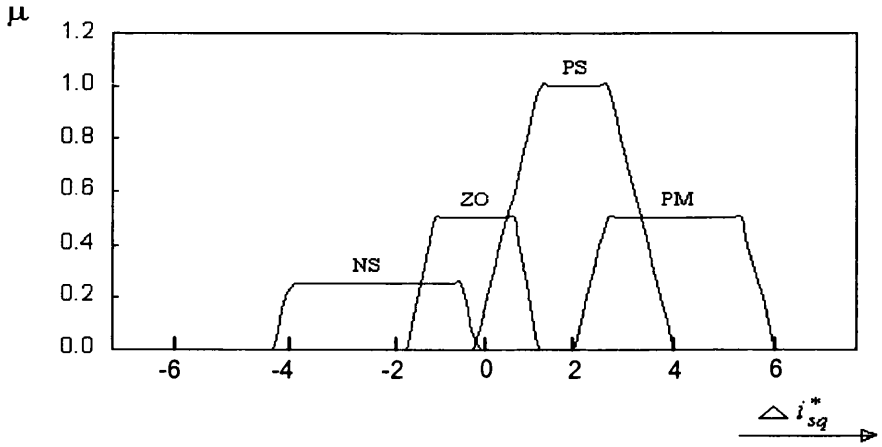


Figure 4-5: The inferred fuzzy control action.

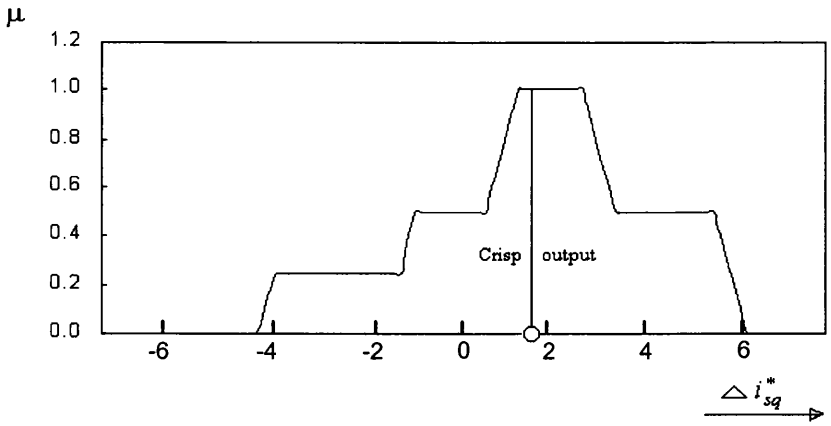


Figure 4-6: The defuzzified output.

4.6 Summary:

In this chapter, a review of recent applications of FLC in ac speed drives has been presented. It is followed by a brief summary of some of the relevant concepts in fuzzy set theory and fuzzy logic. The main ideas underlying the FLC have been reviewed. The concept of solving any control problem by a set of linguistic rules has been highlighted. The reminder of the chapter then focused on the procedures to build a simple fuzzy controller to regulate the speed of the induction motor. The behaviour of the designed controller has been tested via simulation in MATLAB and SIMULINK. The following are the major points covered in this chapter:

1. Fuzzy logic control is mainly based on a logical model, which simulates the thinking process of a human operator while controlling the system manually. It usually requires no mathematical model of the controlled system in its design but a graphical representation of the system behaviour is essential.

2. The performance of the fuzzy controller is characterised by a set of linguistic rules that are developed according to the expert knowledge of the controlled system. In most engineering control applications, these rules are simply expressed in the form of "IF-THEN" conditional statements.

3. The computer model developed in MATLAB and SIMULINK in this chapter will be then used to design and build a fuzzy controller for variable speed drive system as demonstrated in Chapter 6.

Chapter 5, Application of variable structure control with sliding mode:

5.1 Introduction:

In the previous chapter, a detailed review of relevant concepts of Fuzzy Logic Control (FLC) as one of the modern control techniques has been presented. The application of FLC in ac speed drive systems as well as design procedures have been also reviewed. In this chapter, Variable Structure Control with Sliding Mode (VSC-SLM) as another example of modern control algorithms is highlighted. The core role of this control technique is to force the state trajectory of the controlled system to a pre-determined manifold surface, known as the *sliding surface.* The control law is then designed so that the system trajectory always reaches the sliding surface. This is known as the *reaching mode* (Gao and Hung 1993; Hung et al. 1993). Once on the sliding surface, the control structure is changed discontinuously to maintain the system trajectory sliding on the surface towards the origin. At this stage the system is in the *sliding mode.*

The theory of VSC-SLM was first launched in the 1950's. Since then it has been further developed and received worldwide recognition for controlling electrical drives (Ho and Sen 1991; Hung et al. 1993). It has been reported in literature that it is capable of providing a fast dynamic response insensitive to system parameter variation and external disturbances. However, chattering phenomenon about the sliding surface is inevitably generated with this type of control scheme. This is due to the delay of the control signal, which results in low control accuracy. If this problem persists, it might excite un-modelled dynamics such as actuators and sensors.

In recent years, many researchers have investigated the possibility of minimising the chattering effects in the VSC-SLM control structure (Chang et al. 1990; Ho and Sen 1991; Gao and Hung 1993; Ishigame et al. 1993; Utkin 1993; Kawamura et al. 1994). In this work, a new VSC-SLM control design using concepts from fuzzy logic algorithm is proposed to alleviate the chattering problems and enhance the dynamic response of the ac drive system. The essence of the new VSC-SLM control algorithm resides in its generalised form of switching behaviour, which allows a smooth transition between the reaching mode and the sliding mode.

In this chapter, a brief review of recent applications of VSC-SLM in speed drive systems is presented. The basic theory and general mathematical representation of VSC-SLM are considered in brief. This is followed by a design procedure and stability analysis of the conventional VSC-SLM control structure. Finally, the development of the new integrated Fuzzy Sliding Mode Control (FSLMC) scheme that is based on the concept of "*reaching law*" is thoroughly discussed.

5.2 Application of VSC-SLM in speed drive systems:

In many speed drive applications, it becomes a must to develop a control system that is properly designed to be insensitive to parameters variation and load disturbances. This can be achieved with the use of modern control techniques such as FLC and VSC-SLM control schemes. In the past few years, the latter type of control scheme has been successfully applied in versatile control applications, in which robustness against parametric variations and external disturbances is ensured. The main features of VSC-SLM controller are (1) high speed of response, (2) good transient response, (3) simplicity of design and implementation, and (4) insensitivity to changes in system parameters. Therefore it is usefully employed in systems with uncertain and time-varying parameters such as speed drive systems. In what follows, a summary of recent applications of VSC-SLM controller in drive systems is reviewed.

The early use of VSC-SLM in the application of dc motor drives has been presented by Lin and Tsai (1984). In their work, they have applied the principle of VSC-SLM to an incremental servo system, in which the position of the dc motor is being controlled. According to some experimental results, it has been found that VSC-SLM exhibits fast dynamic response without overshoot compared to a conventional control system whose structure is fixed.

The implication of VSC-SLM in speed regulation of dc drives was considered by Lim et al.(1991), Utkin (1993), Chung et al. (1994), and Hung et al. (1994). Lim et al. (1991) have presented a new approach to design a microprocessor based VSC-SLM controller for a brushless dc motor drive. In addition, a modified design procedure has been proposed for discrete time VSC-SLM design. The new design has been verified using simulation and experimental set-up and the results obtained

demonstrated the robust properties of the VSC-SLM controller. Another approach to design a variable structure controller has been proposed by Hung et al. (1994), where the sliding surface equation is synthesised from the feedback signal only. Experimental results revealed the superiority of the proposed VSC-SLM over PI controller when they are both exposed to the same operating conditions. In contrast to dc motors, ac motors (in particular induction type) are extensively used nowadays in many drive applications due to their simpler and more economical construction. But, however, these motor types require a complex control method to overcome the non-linearity and vulnerability to parameter variations in the drive system. As a result, the control of ac drives represents the most popular and challenging applications of VSC-SLM.

The literature is very fruitful with publications concerning the use of VSC-SLM in induction motor (IM) control applications. Sabanovic and Izosimov (1981) presented a novel and rather simple procedure of VSC-SLM synthesis for the control of position, speed and torque of squirrel cage induction motor type. A series of experiments have been performed to justify the approach to the proposed control system synthesis. Another work was presented by Sabanovic and Bilalovoic (1989), in which a unified approach to design VSC-SLM control system for induction, synchronous and dc brushless motors has been proposed. This work revealed that a simple control algorithm, which is derived from the VSC-SLM theory can be used to control any one of the aforementioned motor types with the basic control structure maintained unchanged.

Application of VSC-SLM in the outer loop of a speed drive system utilising a series connected wound rotor induction machine (SCWRIM) was presented by Ho and Sen (1990). In this work, an integral compensation has been used in conjunction with VSC-SLM to reduce the effect of high torque ripples and steady state errors. In addition, constant acceleration sliding line segments have been introduced in the controller design to provide better utilisation of the torque capability of the drive system. It followed from some experimental results that the robustness obtained by the proposed method of control is superior to the conventional method using PI control.

The same authors (Ho and Sen 1991) have presented another work, in which new techniques of robustness improvement for the VSC-SLM drive systems have been proposed. This includes a novel method of control in which the equation of the sliding surface has been feed-forwarded to enhance the

control dynamics. In addition, different acceleration estimation methods such as observers and non-ideal differentiators have been outlined and compared. The effectiveness of the proposed controller has been demonstrated through computer simulations and experimental set-up.

Very recently, Benchaib et al. (1999) presented a performance comparison between the VSC-SLM control, conventional field oriented control (FOC) and the input-output linearisation control (IOLC) techniques. All the three control methods have been successfully implemented on-line in real time on a digital signal processor (DSP), and a comparative study has been carried out. The experimental results showed the superiority of the VSC-SLM compared to the other two controllers with respect to the rotor resistance variation and low speed operation. However, the VSC-SLM has a high-resonant noise level.

5.3 The chattering effects:

The distinctive feature of VSC-SLM in dealing with nonlinearities of control systems has attracted significant attention in research community worldwide. However, one of the major difficulties with VSC-SLM design is the chattering problem about its sliding surface (also known as switching surface). In theory, the control structure of VSC-SLM is switched alternately at infinite switching frequency to provide a smooth control signal. In practical systems, however, it is not possible to achieve infinite switching frequency due to the inherent time delay in the switching devices and the delay caused by the computational time of the control algorithm (Utkin 1993). Thus the switching frequency is confined to a finite value, which is the main cause of the chattering along the sliding surface.

Since the chattering phenomenon is inevitably generated in the conventional VSC-SLM, there exist several methods to eliminate or reduce its effect. One method is called the "*continuous approach*" [87-88], in which a *boundary layer* around the sliding surface has been introduced. The control within this layer is a continuous approximation of the actual discontinuous control signal. Another method is called "*Chattering Alleviation Control*" (CAC), which has been proposed by Chang et al. (1990) to eliminate the chattering problems. In this method two important ingredients have been considered which are the linear state feedback and the conventional VSC-SLM. According to the CAC rules, a combination of both will eliminate the chattering effects completely.

Another approach has been suggested by Utkin (1993) and Kawamura et al. (1994), where chattering can be reduced by a combination of disturbance full-order observer and feed-forward control. The dither signal (Harashima et al. 1985), represents the core role of this control design due to its implication of causing the chattering, and is also responsible for the suppression of the disturbances. This method employs a feed-forward compensation of the disturbance, which is to be estimated by the disturbance observer. However, an observer requires exact knowledge of the plant parameters to yield accurate estimates, so that the robustness of the controlled system is compromised (Utkin 1993).

One method has been proposed by Gao and Hung (1993), Hung et al. (1993), and Ghalia and Alouani 1995), which deals directly with the reaching process before the system hits the sliding surface. This technique is called the "*reaching law method*". It is based on a differential equation that directly specifies the dynamics of the switching function. The control signal is then determined by computing the time derivative of the switching function along the reaching mode trajectory. This method proved robustness of the control system and reduced chattering. However, the designed control signal does depend on perturbation and disturbances and it should include their parameters.

While the research efforts continue in the improvement of VSC-SLM control, some researchers have identified the connection between fuzzy logic and VSC-SLM (Ishigame et al. 1993; Suyitno et al. 1993; Ghalia and Alouani 1995; Liu and Lin 1996; He et al. 1997; Lin et al. 1997; Lin and Chiu 1998). This connection has suggested the integration of the two control approaches in many control system design applications. In this work, a new method of combining both control algorithms has been proposed to reduce the chattering effects inherited in VSC-SLM and improve the dynamic response of the speed drive system.

Furthermore, the insensitivity property of the VSC-SLM is present only when the system states are in the '*sliding mode*'. In other words, the system response is still sensitive to parameter perturbations and external disturbances before the system states hit the sliding surface. To overcome this shortcoming, two control laws have been used in the combined control system with respect to the active mode of operation. The first is based on the '*reaching law*' method, when the system states are in the '*reaching mode*', while the second is based on the direct switching approach when the system states are in the '*sliding mode*'. Both of these modes of operation will be briefly defined in the following sections.

The switching between the two control laws, however, includes fuzziness since it is not easy to determine whether the system states are in the sliding mode or in the reaching mode. Therefore, '*fuzzy set theory*' has been used to allow a smooth transition between the two control laws when the system states switch between the two modes of operation.

5.4 Basic principles of VSC-SLM:

To highlight the basic concepts of variable structure control schemes, a simple second order nonlinear system is considered. This system can by described in the state space plane by the following state equations,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= a\,(x,t) + b\,(x,t)\,u \end{aligned} \tag{5.1}$$

where $dim\_x = n$ and $dim\_u = m$, $\quad n > m$

From equation (5.1), $x = [x_1, x_2]^T$ is the state vector of the system, *a (x,t)* and *b (x,t)* are vector fields that represent the nonlinear functions, and u is the control input to the system. In theory, the VSC-SLM implies the design of a sliding surface $s(x)$ of an appropriate m-dimensional, which is of a lower order than the given plant to represent the desired sliding mode dynamics. Thus, the sliding surface can be defined as follows,

$$s(x) = Cx \tag{5.2}$$

where,

$$s(x) = [s_1(x) \quad s_2(x) \ \ldots\ s_m(x)]^T \tag{5.3}$$

and,

$$C = [c_1^T \quad c_2^T \ \ldots\ c_m^T]^T \tag{5.4}$$

therefore,

$$s_i(x) = c_i^T x = 0, \qquad i = 1, 2, \ldots, m \tag{5.5}$$

The general form of the VSC-SLM control law u is described by,

$$u = \begin{cases} u^{+}(x,t) & \text{if } s(x,t) > 0 \\ u^{-}(x,t) & \text{if } s(x,t) < 0 \end{cases} \tag{5.6}$$

It follows from the above equation that the system trajectory is forced to reach the switching surface in a finite time with a small overshoot. Once the system state is on the switching surface it slides towards the origin, causing the system to be in the sliding mode, in which it follows the desired dynamics. The control signal is then designed to satisfy the existence of the sliding mode by satisfying the reaching condition, which is given by

$$s(x)\dot{s}(x) < 0 \tag{5.7}$$

This is known as the '*direct switching function approach*' (Hung et al. 1993). The objective of this approach is to design a controller so that the system follows a prescribed dynamics and becomes insensitive to parameter variation. Furthermore, a fast dynamic response without overshoot is highly desirable. This can be achieved by a good selection of a sliding surface that is defined by,

$$s(x) = Cx_1 + x_2, \qquad C > 0 \tag{5.8}$$

Therefore, the control law based on this approach is of the form,

$$u_S = \varphi_1 x_1 + \varphi_2 x_2 \tag{5.9}$$

where

$$\varphi_1 = \begin{cases} \alpha & \text{if } sx_1 > 0 \\ \beta & \text{if } sx_1 < 0 \end{cases}$$
$$\varphi_2 = \begin{cases} \gamma & \text{if } sx_2 > 0 \\ \xi & \text{if } sx_2 < 0 \end{cases} \tag{5.10}$$

Each gain constant (φ_1 or φ_2) must be selected to satisfy the reaching condition given in (5.7) which makes the VSC-SLM system globally asymptotically stable. To test the stability of the system, '*Liapunov's theorem*' is used. The basic definition of Liapunov function is given by,

$$V = \frac{1}{2} s^2 \tag{5.11}$$

Considering Liapunov's direct method, if V is positive definite and $\dot{V}$ is negative definite then the system state will be driven toward the sliding surface and remain sliding on it until the origin is reached asymptotically (Ogata 1988). Based on this procedure, the reaching condition for the existence of the sliding mode can be derived by,

$$\dot{V} = s\dot{s} < 0 \tag{5.12}$$

which is the same as the reaching condition that is given in (5.7). To further demonstrate the basic principle and derivation of VSC-SLM control, a simple example of second order system is considered (Hung et al. 1993; Utkin 1993; Abed 1995). The block diagram of this system is depicted in Figure 5-1, and the system model is defined by the following transfer function given in Laplace form as,

$$\frac{Y(s)}{R(s)} = \frac{K}{K + S^2} \tag{5.13}$$

and the system states are given by,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -K x_1 \end{aligned} \tag{5.14}$$

It should be noted that the system transfer function given in (5.13) is derived when the system is operating in the negative feedback mode, i.e. the switch SW (as shown in Figure 5-1) is connected to point 'A'. On the other hand, if SW is switched back to point 'B' then the system is said to be operating in the positive feedback mode and the transfer function would be,

$$\frac{Y(s)}{R(s)} = \frac{K}{K - S^2} \tag{5.15}$$

and the system states are given by,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= K x_1 \end{aligned} \tag{5.16}$$

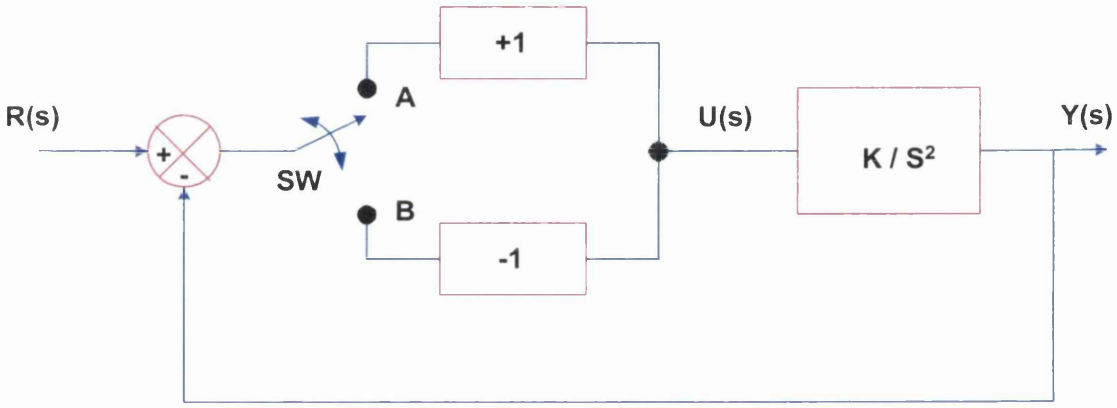
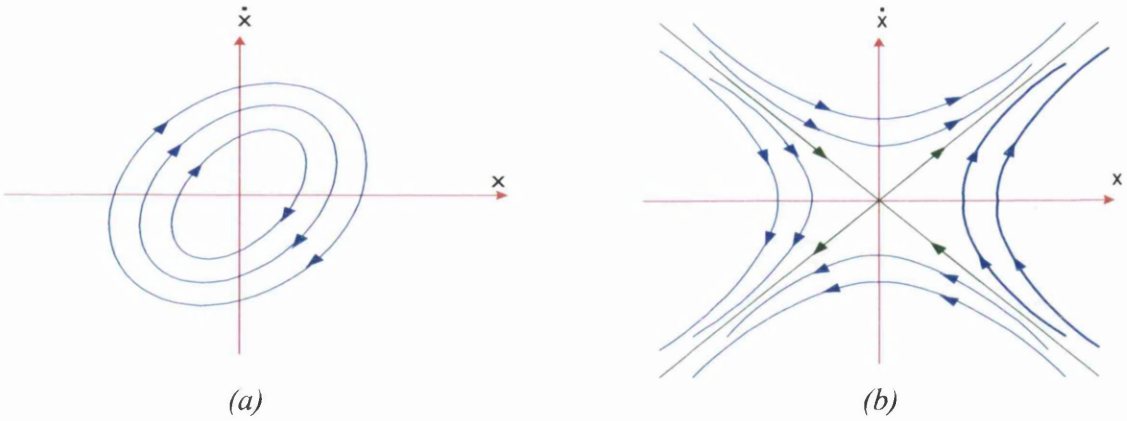


Figure 5-1: Block diagram of the second order system model.

It follows from the above modes of operation that the switch (SW) has been used to alternately change the control structure such that the system is either operating with the negative or positive feedback, hence the term variable structure control. The phase plane trajectories for the second order system in both modes of operation are shown as portraits in Figure 5-2. It should be noted that neither of these two control structures are asymptotically stable. However, asymptotic stability is achieved if the structure of the system is switched between positive and negative feedback along an asymptote line, which is often called the sliding line and it is defined by,

$$s(x) = Cx_1 + x_2, \qquad C > 0 \tag{5.17}$$



(a) *(b)*

Figure 5-2: Phase plane trajectories of second order system
(a) for negative feedback; (b) for positive feedback

Thus, new system properties are obtained to compose the desired trajectory from both control structures. This trajectory describes a new type of motion along the sliding surface at $s(x) = 0$ called a '*sliding mode*'. In other words, the state trajectories of the system are always directed towards the sliding surface by an appropriate choice of feedback structure (negative or positive). Once on the sliding surface, the control structure switches between negative and positive feedback alternately to force the state trajectories to remain sliding along the sliding surface towards the origin, thus earning the name '*sliding mode control*'. The resultant phase portrait of the combined phase trajectories utilising the new sliding motion trajectory is shown in Figure 5-3.

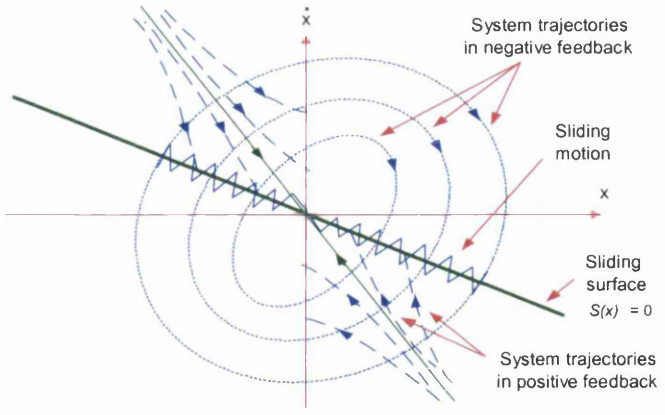


Figure 5-3: Phase plane trajectory of second order system
with sliding mode control

It should be noted that during the sliding mode motion the system dynamics are solely governed by the parameters that describe the sliding surface which are given in Equation (5.17). In addition, the sliding mode is a trajectory that is not inherent in either of the two structures defined by the negative and positive feedback, resulting in an improved performance of the system dynamics. Furthermore, in theory, it is known that the transient dynamics of VSC-SLM systems consists of two parts, representing two modes of operation. The first part is the *reaching mode* (or non-sliding mode), in which the system trajectory moves from anywhere on the phase plane towards the sliding surface at finite time. The second part is the *sliding mode*, in which the system trajectory keeps sliding along the sliding surface towards the origin of the phase plane. Graphical illustrations of the sliding and reaching modes are shown in Figure 5-4.

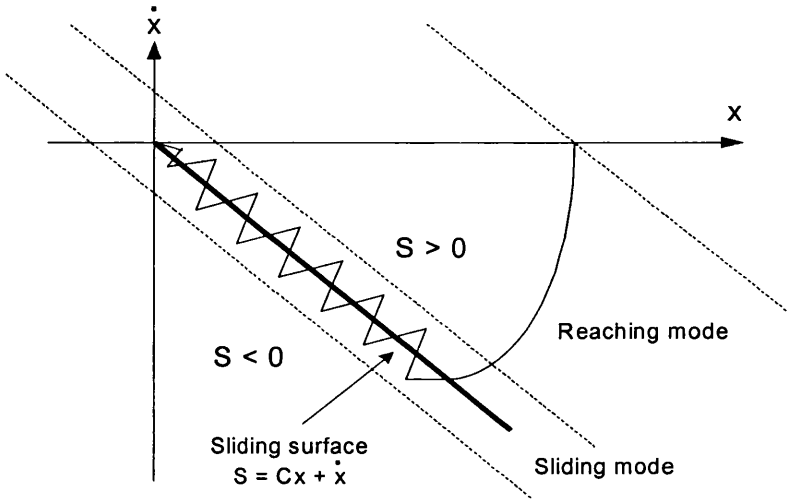


Figure 5-4: Phase plane trajectory with reaching and sliding modes

It is useful to note that the most important virtues of VSC-SLM are its robustness and invariance to system parameters perturbation and external disturbances. In the conventional VSC-SLM control systems, the invariance property is only present once the system states are operating in the sliding mode, and therefore, robustness cannot be ensured throughout the entire transient response. In other words, during the reaching mode the system does not posses the invariance property and is still sensitive to parameters variation and external disturbances (Harashima 1985; Hung et al. 1993). Therefore, it is important to minimise the time required to reach the sliding surface.

5.5 *The proposed synthesis of VSC-SLM using fuzzy logic:*

In most of the previous designs of VSC-SLM systems, the transient dynamics of the reaching mode are not usually defined. Thus, the speed with which the sliding surface is reached and the amount of control chattering around it are not specified. Therefore, an alternative design of VSC-SLM is required to give desired transient dynamics in the reaching mode as well as in the sliding mode. Since, the reaching mode presents an important part of a VSC-SLM system, a new control law has been proposed by Gao and Hung (1993), to control its transient dynamics once the system states are in the reaching mode. This law is often called a '*reaching law*', in which the dynamics of the sliding surface *s(x)* are specified. These dynamics are described by a differential equation of the form,

$$\dot{s}(x) = -K\, sign(s), \qquad K > 0 \tag{5.18}$$

It follows from the above equation that the speed with which the system states approach the sliding surface can be predetermined, where K gives the rate of convergence. Thus, the control signal of the reaching law can be defined by computing the time derivative of *s(x)* along the reaching mode trajectory of system (5.1) which yields,

$$\dot{s}(x) = \frac{d\,s}{d\,x} a(x,t) + \frac{d\,s}{d\,x} b(x,t)\, u = -K\, sign(s) \tag{5.19}$$

Rearranging Equation (5.19) for the reaching control law gives,

$$u_R = \left[\frac{d\,s}{d\,x} b(x,t)\right]^{-1} \left[\frac{d\,s}{d\,x} a(x,t) + K\, sign(s)\right] \tag{5.20}$$

This law forces the system state trajectory to reach the sliding surface *s(x)* at a constant rate defined by the value of control gain K. Selecting a large value of K shortens the reaching time but it would be on the expense of causing severe chattering once the system states hit the sliding surface. In contrast, if a smaller value of control gain K is selected it results in less chattering while on the sliding surface. The reaching time in this case, however, would be long and as a result the system is susceptible to

parameters variation and external disturbance. Therefore, the characteristics of the system dynamics during the reaching mode are still fuzzy. Fuzzy logic concepts can be usefully integrated with the current control design in the reaching mode to regulate the value of control gain *K*, which is solely responsible to control the reaching time. In the light of the above design considerations, the value of *K* has to be selected according to some rules that are based on the position of the system state trajectory with respect to the sliding surface on the phase plane.

This position is actually determined by the value of the sliding surface *s(x)* which is defined in Equation (5.17) at every instant of time *(t)*. For example, a large gain *K* is inferred from the fuzzy logic controller when the state trajectory is far away form the sliding surface, i.e. *s(x)* is large. Then it decreases gradually as the state trajectory approaches the sliding surface (the value of *s(x)* becomes smaller) in order to reduce the chattering effects.

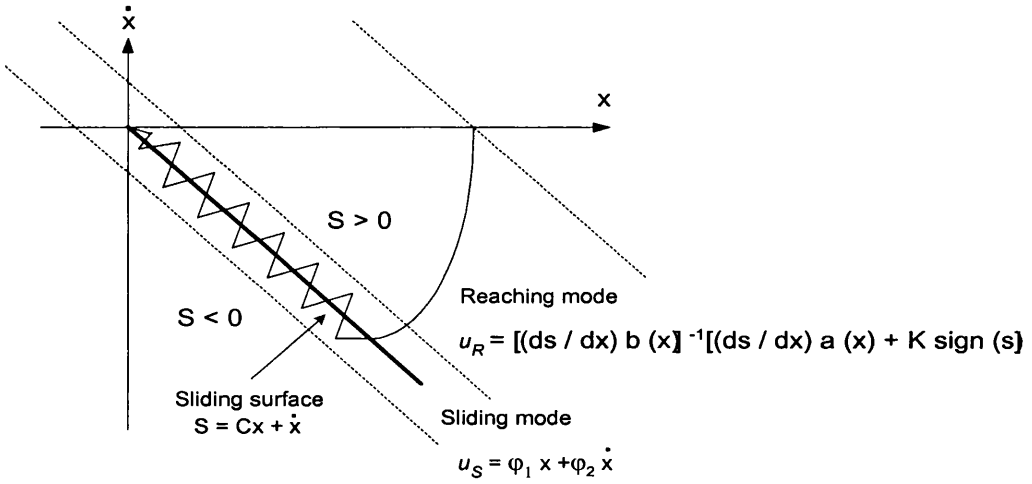


Figure 5-5: Selections of the control laws

In summary, based on the aforementioned control approaches, two control laws can be utilised depending on the position of the state trajectory on the phase plane as depicted in Figure 5-5. The reaching control law u_R with an integrated fuzzy logic control (Equation (5.20)) is used when the state trajectory is in the reaching mode, whereas the control law u_S (Equation (5.9)) is used in the sliding mode. Thus, the invariance property in the entire transient response of the VSC-SLM is well preserved. However, the switching from u_R to u_S includes fuzziness since it is not easy to know whether or not

the state trajectory is in the sliding mode (Ishigame et al. 1993; Ghalia and Alouani 1995). Fuzzy set theory is then applied to the two control laws to allow a smooth switching between them, in order to maintain the chattering effects within an acceptable level.

5.6 Design methods of the new integrated control scheme for an ac speed drive:

In this section, the design methods for the proposed control system to regulate the rotor speed of the induction motor are presented. This system is then used in the following chapter to assess its dynamic performance by simulation when it is applied to the outer-speed loop of an ac speed drive. In general, the control objectives for a high performance speed drives are: 1) quick tracking performance of the reference speed changes without overshoot, and 2) the maximum speed drop and the restore time when load is applied must be kept as small as possible. In the previous chapter, a simple FLC controller has been designed to accomplish these objectives. Here, a proper design of the proposed controller is highly desirable to meet the same objectives as well as to improve upon the dynamic characteristics of the drive system. The following steps briefly highlight the general design procedures for the proposed control scheme.

Step 1: Define the mathematical model of the controlled system:

In order to construct a VSC-SLM controller, the mathematical model of the controlled system has to be defined. This model is significantly used to design the reaching and sliding control laws. For example, the control objective of an ac speed drive system is to make one of the mechanical co-ordinates such as the rotor speed be equal to the reference input speed. Thus, the complex model of an ac machine can be reduced to a first order differential equation that only defines the system mechanical motion. In addition, it would be more convenient to express this dynamic equation in the state space representation, in which the system states are defined by the speed error and its derivative (Ho and Sen 1990, 1991). A simplified electromechanical model of a speed drive system together with an outer-loop speed control is illustrated in Figure 5-6.

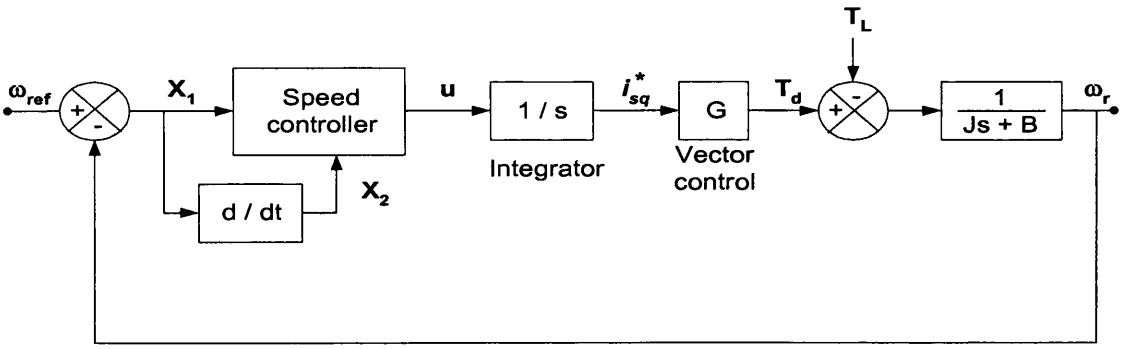


Figure 5-6: Electromechanical model of speed drive system.

The motion governing equation of the induction motor (IM) drive in time domain is given by,

$$\frac{d\omega_r}{dt} = \frac{1}{J}(T_d - T_L - B\omega_r) \tag{5.21}$$

This equation can be rewritten in the S-domain as,

$$s\omega_r(s) = \frac{1}{J}(T_d(s) - T_L(s) - B\omega_r(s)) \tag{5.22}$$

and,

$$T_d(s) = \frac{G\,u(s)}{s} \tag{5.23}$$

where *u(s)* is referred to the controller output, while *G* represents the control regulator of the induction motor such as the scalar or vector control. Substituting Equation (5.23), into (5.22) and rearranging gives,

$$s^2 J\omega_r(s) + s\,B\omega_r(s) = G\,u(s) - s\,T_L(s) \tag{5.24}$$

This equation can be rewritten in the time domain to yield the following,

$$\frac{d^2\omega_r}{dt^2} = -\frac{B}{J}\frac{d\omega_r}{dt} + \frac{G}{J}u \tag{5.25}$$

The term T_L disappears from the above equation because it is a constant value. For the proposed control scheme the speed error and its derivative are selected to be the system state variables, which are given by,

$$x_1 = \omega_{ref} - \omega_r \tag{5.26}$$

and,

$$x_2 = \frac{d\,\omega_{ref}}{dt} - \frac{d\,\omega_r}{dt} = -\frac{d\,\omega_r}{dt} \tag{5.27}$$

Therefore,

$$\dot{x}_1 = x_2 \tag{5.28}$$

and,

$$\dot{x}_2 = -\frac{d^2\omega_r}{dt^2} \tag{5.29}$$

Substituting Equation (5.25) into (5.29) gives,

$$\dot{x}_2 = +\frac{B}{J}\frac{d\,\omega_r}{dt} - \frac{G}{J}u \tag{5.30}$$

or

$$\dot{x}_2 = -\frac{B}{J}x_2 - \frac{G}{J}u \tag{5.31}$$

From the above formulations, the state space representation of the speed drive system is given as,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -B/J \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -G/J \end{bmatrix} [u] \tag{5.32}$$

This matrix equation implies a good design of VSC-SLM control system to generate a control signal u, so that the system states (x_1, x_2) follow a prescribed dynamics regardless of the parameters perturbation of the drive system. It also requires a proper selection of a sliding surface, where the system trajectory slides on towards the origin.

Step 2 Selection of the sliding surface:

The essence of the VSC-SLM design lies in the proper selection of the sliding surface in which it represents the desired system dynamics in the sliding mode. The general form of this surface is defined in Equation (5.17), where a sufficient value of C is chosen to obtain a desired speed response. It is useful to note that the sliding surface is always of lower order than the dimension of the given controlled plant. For instance, the sliding surface of a simple second order system having only one input is a straight line equation in which the slop edge is controlled by the value of C (Harashima et al. 1985). In theory, a fast dynamic response can be achieved if a steep sliding surface is used. Therefore, a compromise has to be made in selecting a sufficient value of slope constant to achieve a quick transient response with a minute chattering effects. In this design, the coefficient C has been selected to be equal to 4.2.

Step 3: Design of VSC-SLM control laws:

Having developed the system model as well as the design of the sliding surface, the next step is to design the control laws that give the desired dynamics in the reaching and sliding modes. The core objective of this control is to bring the system states from any position on the phase plane to the sliding surface by an appropriate choice of control gains. The main requirement, however, is that the control should satisfy the reaching condition given in Equation (5.7), which in turn guarantees the existence of the sliding mode on the sliding surface. Additional requirements also include fast reaching response, minute chattering effects and robustness restraint during the entire transient response.

1) Reaching mode control signal: The reaching law is defined by the following differential equation which specifies the dynamics of the sliding surface $s(x)$ as,

$$\dot{s}(x) = -K\, sign(s), \qquad K > 0 \tag{5.33}$$

The control signal based on this approach is determined by computing the time derivative of *s(x)* along the reaching mode trajectory which yields,

$$u_R = \left[\frac{d\,s}{d\,x} b(x,t)\right]^{-1}\left[\frac{d\,s}{d\,x} a(x,t) + K\ sign(s)\right] \tag{5.34}$$

Taking the time derivative of the sliding surface equation which is defined in (5.17) gives,

$$\dot{s} = C\ \dot{x}_1 + \dot{x}_2 = C\,x_2 + \dot{x}_2 = -K\ sign(s) \tag{5.35}$$

Substituting Equation (5.31) into (5.35) yields,

$$C\,x_2 - \frac{B}{J}x_2 - \frac{G}{J}u = -K\ sign(s) \tag{5.36}$$

Rearranging the above equation and solving for the reaching control signal gives,

$$u_R = \frac{J}{G}\left[C\,x_2 + K\ sign(s)\right] - \frac{B}{G}x_2 \tag{5.37}$$

<u>*2) Sliding mode control signal:*</u> The control signal that is used to force the state trajectory to remain sliding on the sliding surface at $s(x) = 0$, is of the form,

$$u_S = \varphi_1 x_1 + \varphi_2 x_2 \tag{5.38}$$

where the coefficients φ_1 and φ_2 are the controller gains that are selected to ensure the system states move along the sliding line towards the origin point at which the error is zero. These gains are changed discontinuously such that,

$$\varphi_1 = \begin{cases} \alpha & \text{if } sx_1 > 0 \\ \beta & \text{if } sx_1 < 0 \end{cases}, \quad \varphi_2 = \begin{cases} \gamma & \text{if } sx_2 > 0 \\ \xi & \text{if } sx_2 < 0 \end{cases} \tag{5.39}$$

It follows from the gain selections given in (5.39) that the controller gains α, β, γ and ξ, are derived to mainly satisfy the existing condition of the sliding motion, which is given in Equation (5.7). The value limits of these gains can be defined by first computing the time derivative of the sliding line equation that gives,

$$\dot{s} = C\dot{x}_1 + \dot{x}_2 = Cx_2 + \dot{x}_2 \tag{5.40}$$

which can be written as,

$$\dot{s} = Cx_2 - \frac{B}{J}x_2 - \frac{G}{J}u \tag{5.41}$$

Substituting Equation (5.38) into (5.41) gives,

$$\dot{s} = -\frac{G}{J}\varphi_1 x_1 + \left(C - \frac{B}{J} - \frac{G}{J}\varphi_2\right)x_2 \tag{5.42}$$

Therefore,

$$s\dot{s} = -\frac{G}{J}\varphi_1 x_1 s + \left(C - \frac{B}{J} - \frac{G}{J}\varphi_2\right)x_2 s \tag{5.43}$$

It follows from the above equation that the existence condition can be satisfied according to the following conditions,

1) if $sx_1 > 0$, then $\alpha > 0$

2) if $sx_1 < 0$, then $\beta < 0$

3) if $sx_2 > 0$, then $C - \frac{B}{J} - \frac{G}{J}\gamma < 0$, and hence $\gamma > \max\left(\frac{CJ - B}{G}\right)$

4) if $sx_2 < 0$, then $C - \frac{B}{J} - \frac{G}{J}\xi > 0$, and hence $\xi < \min\left(\frac{CJ - B}{G}\right)$

Step 4: Synthesis of fuzzy logic algorithms:

Having designed the basic structure of the proposed control scheme via the design of the sliding surface and the control laws, the next step is to construct the fuzzy logic algorithms (FLC) to overcome the ill-defined characteristics of the system dynamics in the reaching mode. The main ideas underlying the FLC and the concepts of solving control problems by a set of linguistic rules have been thoroughly reviewed in the previous chapter. In the proposed FLC, the process variable to be observed and the output control variable are chosen to be the sliding surface $s(x)$ and the control gain K, respectively. This design also includes the derivation of the fuzzy linguistic rules and the construction of the membership functions.

In the light of the design requirements, the fuzzy rules are based on the algebraic value of the switching surface $s(x)$ that is computed by Equation (5.17) at every instant of time (t), and thereby the position of the system states in the phase plane is determined. The designed linguistic rules that are used in the rule-base of the proposed FLC are as follows

if $s(x(t))$ *is* VL Then $K(t)$ *is* KVL
if $s(x(t))$ *is* L Then $K(t)$ *is* KL
if $s(x(t))$ *is* M Then $K(t)$ *is* KM
if $s(x(t))$ *is* S Then $K(t)$ *is* KS
if $s(x(t))$ *is* VS Then $K(t)$ *is* KVS
if $s(x(t))$ *is* Z Then $K(t)$ *is* KZ

where VL is Very Large, L is Large, M is Medium, S is Small, VS is Very Small and Z is Zero. These are the fuzzy sets for the sliding surface $s(x)$. Similarly, KVL (K-Very Large), KL (K-Large), KM (K-Medium), KS (K-Small), KVS (K-Very Small), and KZ (K-Zero)are fuzzy sets for the control gain K.

Following the construction of the linguistic rules, the membership function for each fuzzy set has to be defined. This can be achieved by assigning the grade of membership to each element of the corresponding universe of discourse. The membership functions of triangular type are used to express the fuzzy control variables $s(x)$ and gain K, respectively as shown in Figure 5-7. The Centre of Area

defuzzification method is then applied on the inferred fuzzy action to determine the final numerical value of the control gain *K*.

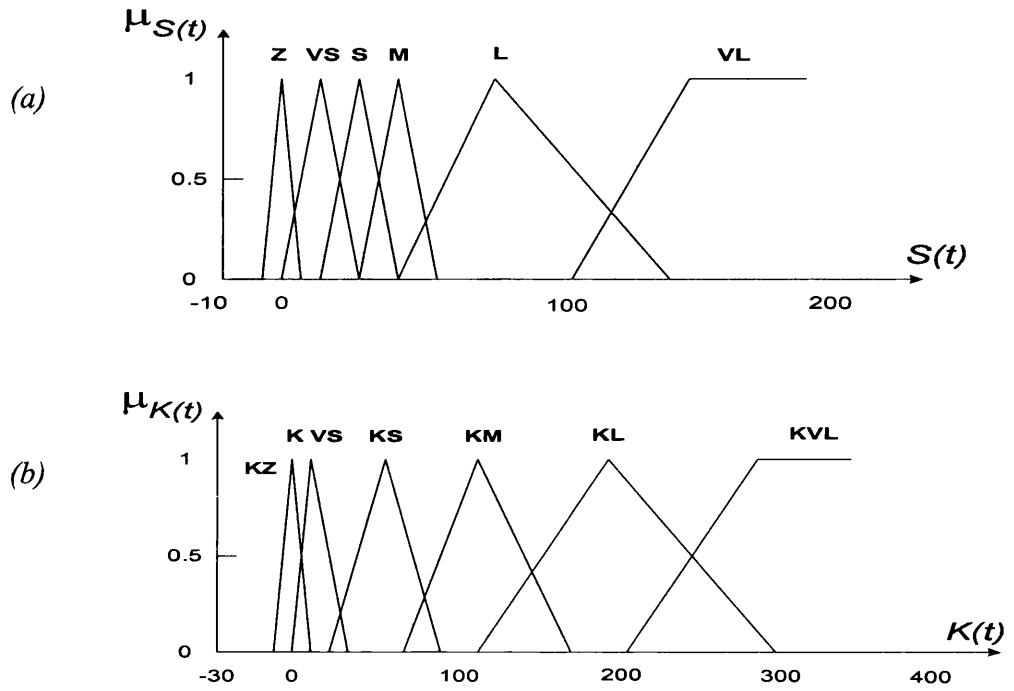


Figure 5-7: Membership functions of: (a) sliding surface s(x), and (b) control gain K

Step 5: Derivation of the overall control law:

After computing the two control signals (u_R and u_S) as described above, the overall control law is then derived to guarantee a smooth transition when the system changes from the reaching mode to the sliding mode. This implies the use of the fuzzy set theory to combine the results of the computed control signals. This can be simply achieved by the proper design of two fuzzy linguistic rules regarding the value of the sliding surface *s(x)* as the condition variable of the form,

Rule 1: *if s(x(t)) is R Then u is* u_R

Rule 2: *if s(x(t)) is S Then u is* u_S

The above rules indicate that if the system states are in the reaching mode (R) then the reaching control u_R is applied to the system. In contrast, if the system states are in the sliding mode (S) then the sliding mode control u_S is applied to the system. Fuzzy inference method is then applied to the switching of u_R and u_S by regarding the value of $s(x(t))$ as the condition variable in the above control rules. The fuzzy reaching and the fuzzy sliding membership functions denoted by μ_r and μ_s respectively are shown in Figure 5.8. The overall control signal can be computed as,

$$u_F = \frac{\mu_r u_R + \mu_s u_S}{\mu_r + \mu_s} \tag{5.44}$$

From the above equation, as the system trajectory gets closer to the switching surface, u_R switches gradually to u_S. Referring to Figure 5-8, when the system trajectory is away from the switching surface (s (x) >= 20 or s (x) <= -20) then $\mu_r = 1$ and $\mu_s = 0$, and therefore only u_R is active and contributes to the final control u_F. However, both u_R and u_S can be active if the system trajectory falls in a mid-point (e.g., 0 < s (x) < 20). As the system trajectory gets closer to the switching surface, the smaller μ_r and the larger μ_s becomes. In other words, the output control u_F is mainly contributed by u_S. If the system remains on the switching surface, the u_R is disabled completely and the system is totally under the traditional sliding mode control u_S.

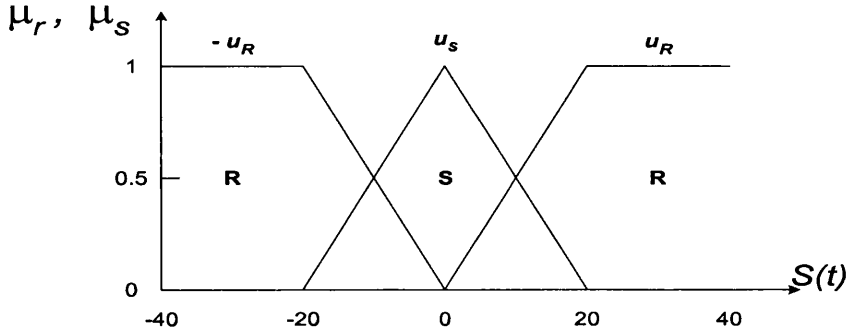


Figure 5-8: Fuzzy reaching and sliding membership functions.

5.7 Summary:

In this chapter, a brief review of recent applications of Variable Structure Control with Sliding Mode (VSC-SLM) in speed drive systems was presented. The basic theory and general mathematical representation of VSC-SLM were briefly considered. This was followed by design procedures and stability analysis of the conventional VSC-SLM control structure. A new integrated Fuzzy Sliding Mode Control (FSLMC) scheme, which is based on the concept of *reaching law* was thoroughly discussed. The following are the main points included in this chapter:

1. VSC-SLM controller is usefully employed in systems with uncertain and time-varying parameters such as ac drives to provide fast dynamic response insensitive to system parameters variation and external disturbances.

2. One of the major difficulties with VSC-SLM controller design is the chattering problem about its sliding surface. This is due to the inherent time delay in the switching devices and the delay caused by the computational time of the control algorithms.

3. The transient dynamics of VSC-SLM systems consists of two parts, representing two modes of operation. The first part is the *reaching mode* (or non-sliding mode), in which the system trajectory moves from anywhere on the phase plane towards the sliding surface at finite time. The second part is the *sliding mode*, in which the system trajectory keeps sliding along the sliding surface towards the origin of the phase plane.

4. In conventional VSC-SLM controllers, the invariance property is only present once the system states are operating in the sliding mode, and therefore, robustness cannot be ensured throughout the entire transient response. In other words, during the reaching mode the system does not posses the invariance property and is still sensitive to parameters variation and external disturbances.

5. The new VSC-SLM controller (FSLMC) is designed using concepts from fuzzy logic algorithms to alleviate the chattering problems, enhance the dynamic response of the ac drives, and ensure control robustness throughout the entire transient response (reaching and sliding regions).

6. Both controllers, the conventional VSC-SLM and the proposed FSLMC, will be designed and tested via simulation for variable speed induction motor drives as shown in Chapter 6.

Chapter 6, Control methods of an ac induction motor:

6.1 Introduction:

As explained in Chapter 3, the dynamics of an induction motor can be represented by a fifth order state space equations, which represent a non-linear multivariable control system. Here, the control inputs are usually the voltages and the frequency and the outputs can be speed, position, torque, stator currents or a combination of them. Although the induction motor, particularly the squirrel cage type, is more rugged and less expensive when compared to the dc motor, the cost of conversion and hardware control equipment is generally high which makes the ac drive system more expensive than the dc drive system. With the advent of the integrated converter and microcomputer-based controllers, however, the cost of control electronic has been reduced significantly. In addition, the microcomputer-based control system solely provides a simplified hardware circuitry, improvement of reliability as well as performance optimisation.

It should be noted that the ac motor requires more complex control schemes than the dc motor, because of its highly non-linear dynamic structure with strong dynamic interactions. The reason for this complexity is due to the inherent dynamic coupling between the direct and quadrature axes (Bose 1882, 1988). Various control techniques of varying degree of complexity have made possible the usage of induction motor in high performance applications. The selection of a particular technique depends on the nature of the application. The most widely accepted control techniques in ac drive systems are the '*scalar control*' (*V/f*) and the '*vector control*' (also known as field oriented control).

In this chapter, the general control concepts regarding the scalar and vector control are briefly reviewed. This is followed by a short discussion of field oriented control techniques with the aid of the transient analysis of the induction motor that was presented in Chapter 3. The dynamic characteristics of the induction motor using various control algorithms with the above control techniques for the speed regulation are assessed via computer modelling and simulation. The control algorithms used in here are the fuzzy logic control, the variable structure control with sliding mode and the proposed control scheme that introduces the fuzzy logic theory into the design of the sliding mode algorithms.

6.2 *Scalar control of an induction motor:*

Scalar control is a term used to include all general non-vector control schemes, in which the control variables are dc quantities and only their magnitudes are controlled. There exist several scalar control methods to regulate the speed and torque of the induction motor. The Volts/Hertz control scheme (also known as *V/f*) using slip regulation is the most widely used in variable speed induction motor drive systems (Bose 1988; Murphy and Turnbull 1988). In this scheme, the stator voltage is varied with the frequency at a constant rate through a phase-controlled PWM bridge inverter, provided that the slip does not exceed the beakdown value (Murphy and Turnbull 1988; Rashid 1993). Such a scheme is characterised by its simplicity and its capability in providing satisfactory steady state behaviour of the induction motor in both open loop and closed loop control.

Based on the steady state analysis presented in Chapter 3, if a sinusoidal phase voltage, $V_s \sin(\omega_s t)$ is applied to the stator windings of the induction motor a rotating magnetic field with a flux magnitude of ϕ_g is produced in the air-gap. This rotating flux wave, in turn, induces an emf, E_g, in the stator winding, which is less than the applied voltage V_s due to the voltage drop across the stator leakage impedance. The induced emf is directly proportional to the rotating flux and the supply frequency, and the rms air-gap flux can be defined as,

$$E_g = K_g \omega_s \phi_g$$

or

$$\phi_g = \left(\frac{E_g}{\omega_s}\right)\frac{1}{K_g} \tag{6.1}$$

where K_g is a constant and depends on the winding factor and the number of series turns per stator phase. It follows from Equation (6.1) that the air-gap flux ϕ_g can be regulated, if the induced voltage and the supply frequency vary simultaneously at a constant ratio E_g / ω_s. In other words, the air-gap flux is nearly constant when the ratio E_g / ω_s has a fixed ratio. From the steady state equivalent circuit of Figure 3-2, the rms rotor current can be defined as,

$$I_r = \frac{E_g}{\sqrt{(R_r / s)^2 + (\omega_s L_{lr})^2}} \tag{6.2}$$

The general torque equation of the induction motor is given by the equation,

$$T_d = \frac{3P}{\omega_s} I_r^2 \frac{R_r}{S} \tag{6.3}$$

Substituting I_r from Equation (6.2) into Equation (6.3) yields,

$$T_d = \frac{3P}{\omega_s} \frac{S E_g^2 R_r}{R_r^2 + (S\omega_s L_{lr})^2} \tag{6.4}$$

Since $S\omega_s = \omega_{sl}$, therefore Equation (6.4) becomes,

$$T_d = 3P \left(\frac{E_g}{\omega_s}\right)^2 \frac{\omega_{sl} R_r}{R_r^2 + (\omega_{sl} L_{lr})^2} \tag{6.5}$$

Using Equation (6.1), the torque equation in (6.5) results in,

$$T_d = 3P(K_g \phi_g)^2 \frac{\omega_{sl} R_r}{R_r^2 + (\omega_{sl} L_{lr})^2} \tag{6.6}$$

It follows from the above equation that the electromagentic torque T_d is directly proportional to the square of the air-gap flux at a given slip frequency ω_{sl}. As a result, if the air-gap flux is kept constant under all operating conditions by a fixed E_g / ω_{sm} ratio, then the torque T_d can be maintained nearly constant over the whole range of operation, thus earning the name Volts/Hertz control. Effectively, the speed-torque curve shifts along the speed axis allowing the drive to operate at various speeds while maintaining the same torque level.

In most applications of *V/f* control scheme, the voltage drop across the stator leakage impedance $(R_s + jX_r)$ is considered to be small in contrast to the supply voltage V_s. Consequently, V_s is considered to be nearly equal to E_g, and instead of E_g / ω_s, the ratio V_s / ω_s is kept constant. At low supply frequencies, however, the voltage drop across the leakage impedance becomes comparable to V_s and the approximation $V_s \cong E_g$ is no longer valid. In this case, the developed torque decreases and the efficiency of the motor is reduced (Gastli and Matsui 1992). Therefore, a sufficient voltage boost is required at low frequencies in order to compensate for the stator impedance drop and maintain a constant torque level. The degree of the voltage boost depends significantly on the load requirements and in some commercial *V/f* drives the user can adjust it manually.

It should be noted that the regulation of the supply voltage and frequency is utilised in the PWM bridge inverter with the use of voltage and frequency modulation techniques as presented in Chapter 2. It has been shown that variable output voltage can be obtained by regulating the modulation ratio, M_r, while the frequency is controlled by varying the reference frequency of the modulating signal. Although the core objective of *V/f* control scheme is to maintain a constant toque throughout the speed range up to the base speed of the drive system, the PWM inverter can still deliver a constant maximum voltage (maximum modulation ratio) at higher frequencies above the rated frequency. Consequently, the induction motor can operate beyond the rated speed (field-weakening region), but with a reduced flux and torque production (Murphy and Turnbull 1988).

A closed-loop speed control of an induction motor using *V/f* with slip regulation is considered here to demonstrate the design objectives of the speed controllers. Figure 6-1 shows a simplified block diagram of such a speed control scheme with a speed feedback loop as well as a shaft encoder. The reference speed denoted by $\omega_{ref}(t)$ is compared with the feedback rotor speed signal $\omega_r(t)$ at each time interval t to determine the speed error signal, $e(t)$. This signal is then passed through a speed controller such as FLC or VSC-SLM to generate a slip frequency command, $\omega_{sl}^*(t)$ which is proportional to the electromagnetic torque as given in Equation (6.6), and hence the developed torque is controlled. The stator frequency command (inverter frequency) $\omega_s^*(t)$ is then determined by adding $\omega_{sl}^*(t)$ to the measured rotational frequency $\omega_{fr}(t)$. The desired inverter voltage V_s^* is defined from

$\omega_s^*(t)$ by a function generator to maintain a constant air-gap flux level in the induction motor. The function generator usually includes a voltage boost to compensate for the gradual appearance of the stator impedance voltage drop at lower stator frequencies.

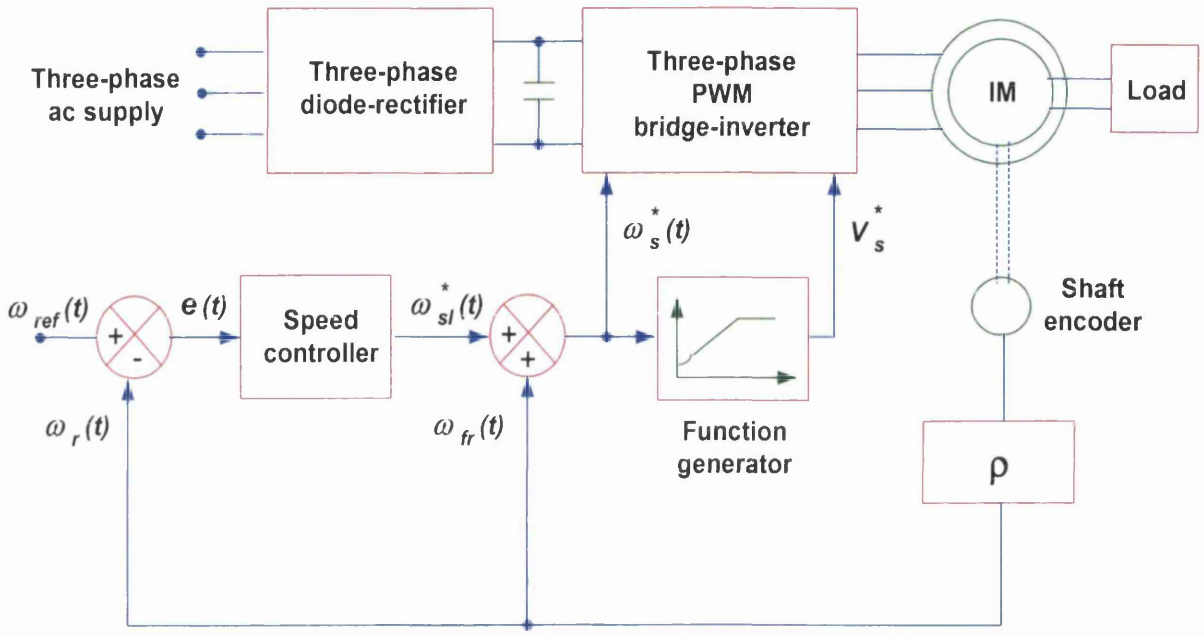


Figure 6-1: Simplified schematics of constant V/f speed control with slip regulation.

In the light of the above control configuration, the speed controller increases the slip frequency command $\omega_{sl}^*(t)$ to a value below the breakdown frequency when a sudden increase in reference speed is demanded. Consequently, the drive accelerates to the set speed with a clamped value of slip frequency which might corresponds to the maximum developed torque.

The slip frequency then drops to a steady state value that is dictated by the value of the load torque. In contrast, if reduction in reference speed is desired, the slip frequency command becomes negative, causing the drive system to operate in the regenerative mode. In this case, a dynamic breaking control circuit is used, in which the regenerated energy due to the drive deceleration is dissipated in a dynamic breaking resistor.

6.3 Vector control of an induction motor:

In the scalar control methods for induction motors, the motor model is considered just for a precise steady state operation to provide a satisfactory steady state performance as presented in the previous section. Therefore, using this type of control strategy, it is expected that the best dynamic performance of the induction motor can not be achieved during the transient stage. This is due to the highly non-linear coupled characteristics of the induction motor. Thus, scalar methods are bound to be unsuitable for induction motor control when utilised in high-performance applications (Bose 1982; Murphy and Turnbull 1988; Hubert 1991).

An improved control system can be designed to de-couple the control of the two components of the stator current. One component i_{sd} provides the air-gap flux, while the other i_{sq} produces the electromagnetic torque. Thus, an independent control of flux and torque is achieved, which is similar to the control principles of a dc machine. Such control method builds upon the good steady state performance obtained with the scalar control, and can give excellent dynamic characteristics in the transient period. The underlying principle of this improved method is known as vector or field oriented control (FOC). It is also called transvector control, because its implementation requires vector transformation from rotating to stationary reference frame, and vice versa.

For the last two decades, enormous number of studies has been carried out on developing vector control techniques and broadening its application in the area of induction motor control. The fundamentals of vector control can be explained with the aid of Figure 6-2. The primary control signals i_{sd} and i_{sq} are converted from field co-ordinates to the corresponding stator co-ordinates. The resulting stationary signals $i_{s\alpha}$ and $i_{s\beta}$ are further converted to the three phase axes (a-b-c) to generate the reference voltage waveforms of the inverter. The inverter circuit is assumed to generate three phase stator voltages as dictated by the corresponding references generated earlier by the controller. These voltages are then converted to the corresponding voltage components in field co-ordinate u_{sd} and u_{sq}, before being applied to the motor model.

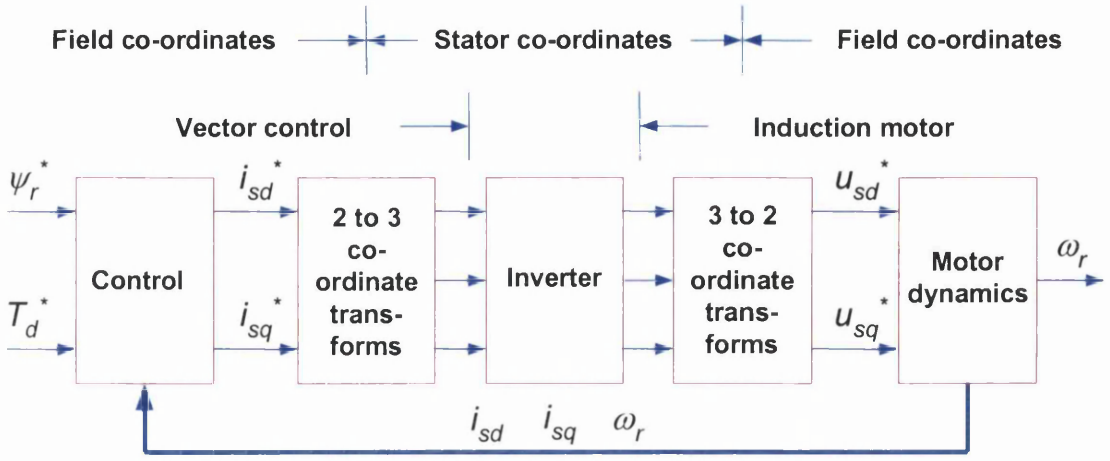


Figure 6-2: Fundamentals of field-oriented control.

There exist several techniques of vector control that depend on how the information of the flux vector is obtained, since it is the basis of co-ordinate transformation (Gabriel et al. 1980; Vas 1990; Liaw and Wang 1991; Miki et al. 1991; Tez 1995; Vas et al. 1997). Also, there are other types of vector control schemes depending on the particular flux vector such as the rotor flux vector, which is used to locate the synchronous reference frame (Vas 1990; Tez 1995; Vas et al. 1997).

Figure 6-3 shows the general classification of vector control methods. The *direct method*, in particular flux sensing method, requires the difficult task of direct measurement of the rotor, stator or air-gap flux amplitude and position. The measured flux is then fed back to the control system for field orientation. The flux can be sensed using flux sensors or stator search coils, but both techniques have the disadvantage that a modified form of induction motor is required, where the sensing coils have to be included in the stator windings.

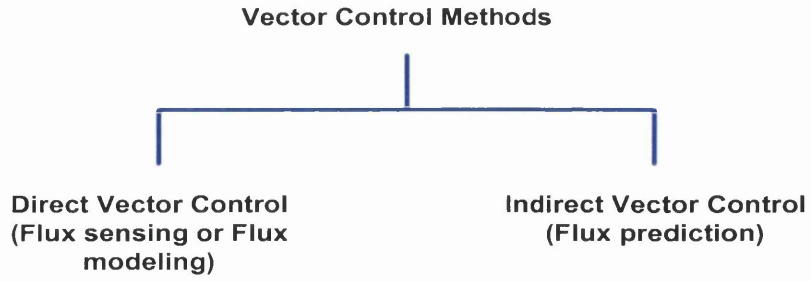


Figure 6-3: General classification of vector control methods.

On the other hand, the flux modelling method is based on the model equations of the induction motor to define the rotor flux vector instead of direct flux measurements. Basically, it uses the slip frequency ω_{sl} to compute the desired flux vector based on a constructed flux model in field co-ordinates. In this scheme, the stator currents and rotor speed are fed back to a rotor flux calculator, which includes a flux model to define the flux position.

In contrast to the direct vector control, the indirect vector control is the most commonly implemented approach in commercial drive systems. This is due to the simplicity of resulting control model, since the direct measuring or modelling of flux are no longer needed as illustrated in Figure 6-4. Nevertheless, the flux position is predicted in a feed forward scheme with the aid of a slip calculator. The output of this slip calculator is the slip frequency, which is then used to partition the stator currents into two components (i_{sd}, i_{sq}).

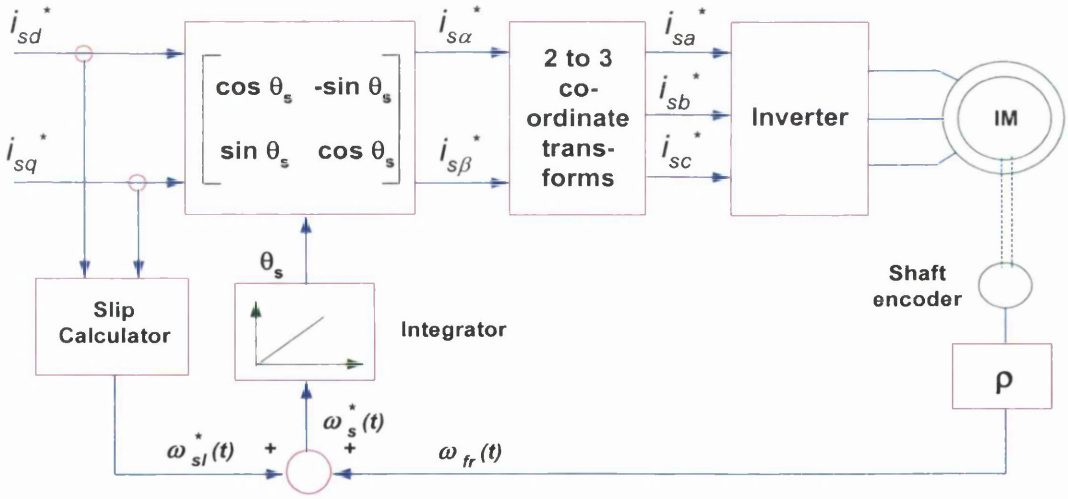


Figure 6-4: Indirect vector control of induction motor.

At this stage, it is important to emphasise on the mathematical approach of field orientation and feed forward slip calculator in order to decouple the control of torque and flux. Recall the rotor voltage vector equation (3.56), derived in the synchronous reference frame,

$$\overline{v_r^f} = R_r \overline{i_r^f} + j\omega_{slip} \overline{\psi_r^f} + \frac{d}{dt} \overline{\psi_r^f} \qquad (6.7)$$

where $\overline{i_r^f}$ and $\overline{\psi_r^f}$ are the space vectors of rotor current and rotor flux linkage, respectively. In the induction motor cage type, the terminal voltage for a short circuit rotor is zero, i.e. $\overline{v_r^f} = 0$. Thus, Equation (6.7) can be rearranged into the following space vector form,

$$\frac{d}{dt}\overline{\psi_r^f} = -R_r\overline{i_r^f} - j\omega_{slip}\overline{\psi_r^f} \tag{6.8}$$

Now, the space vector of the rotor flux linkage given in Equation (3.57) can be rearranged and solved for $\overline{i_r^f}$ as follows,

$$\overline{i_r^f} = \frac{\overline{\psi_r^f} - L_m\overline{i_s^f}}{L_r} \tag{6.9}$$

Substitution of Equation (6.9) into (6.8) and by resolving the flux vector derivative into its two axes components results in the following two equations,

$$\frac{d}{dt}\psi_{rd}^f = -\frac{\psi_{rd}^f}{\tau_r} + \frac{L_m i_{sd}^f}{\tau_r} + \omega_{slip}\psi_{rq}^f \tag{6.10}$$

$$\frac{d}{dt}\psi_{rq}^f = -\frac{\psi_{rq}^f}{\tau_r} + \frac{L_m i_{sq}^f}{\tau_r} - \omega_{slip}\psi_{rd}^f \tag{6.11}$$

where τ_r is the rotor time constant (L_r / R_r). It follows from Equations (6.10) and (6.11) that the rotor flux linkage derivatives are dynamically coupled between the d and q components. However, this coupling can be removed by letting the rotor flux linkage $\overline{\psi_r^f}$ be aligned with the d-axes of the synchronous reference frame, making $\overline{\psi_r^f} = \psi_{rd}^f$ and $\psi_{rq}^f = 0$, hence the term field oriented control (FOC). Thus, Equations (6.10) and (6.11) becomes,

$$\frac{d}{dt}\psi_{rd}^f = -\frac{\psi_{rd}^f}{\tau_r} + \frac{L_m i_{sd}^f}{\tau_r} \tag{6.12}$$

$$0 = \frac{L_m i_{sq}^f}{\tau_r} - \omega_{slip} \psi_{rd}^f \tag{6.13}$$

Therefore, the slip signal ω_{slip} can be computed by rearranging Equation (6.13) as,

$$\omega_{slip} = \frac{L_m i_{sq}^f}{\tau_r \psi_{rd}^f} \tag{6.14}$$

This equation shows that the slip signal is a function of two motor variables, which are the q component of stator current i_{sq}^f and the d component of rotor flux linkage ψ_{rd}^f. For further simplification, if the steady state condition of the motor dynamics is considered, then the rotor flux level becomes constant. Therefore, Equation (6.12) becomes,

$$0 = -\frac{\psi_{rd}^f}{\tau_r} + \frac{L_m i_{sd}^f}{\tau_r} \tag{6.15}$$

and hence,

$$\psi_{rd}^f = L_m i_{sd}^f \tag{6.16}$$

Thus, this equation shows that the magnitude of the rotor flux linkage can be specified by only the d component of stator current i_{sd}^f and hence the cross coupling disappears. Substitution of Equation (6.16) into (6.14) yields,

$$\omega_{slip} = \frac{i_{sq}^f}{\tau_r \, i_{sd}^f} \tag{6.17}$$

On the other side, an alternative torque equation to that in (3.64) is defined as,

$$T_d = \frac{3}{2} P \frac{L_m}{L_r} \left(\psi_{rd}^f i_{sq}^f - \psi_{rq}^f i_{sd}^f \right) \tag{6.18}$$

Using the field orientation method with the rotor flux vector alignment, the above torque equation becomes,

$$T_d = \frac{3}{2} P \frac{L_m}{L_r} \psi_{rd}^f i_{sq}^f \tag{6.19}$$

It follows from this equation that the electromagnetic torque is solely proportional to the algebraic product of the rotor d-axes flux component and the stator q-axes current. However, if the rotor flux level is assumed to be constant, then only the regulation of i_{sq}^f is used to specify the electromagnetic torque production. Moreover, it should be stressed on the de-coupling phenomenon by examining the rotor flux equation in (6.16) as well as Equation (6.19).

According to these equations, the production of flux and torque are de-coupled, and thus any changes in i_{sq}^f will not disturb the flux and an instantaneous torque control can be attained (Heber et al. 1997). Figure 6-5 shows the block diagram of the indirect field oriented control for induction motor drive with speed control loop.

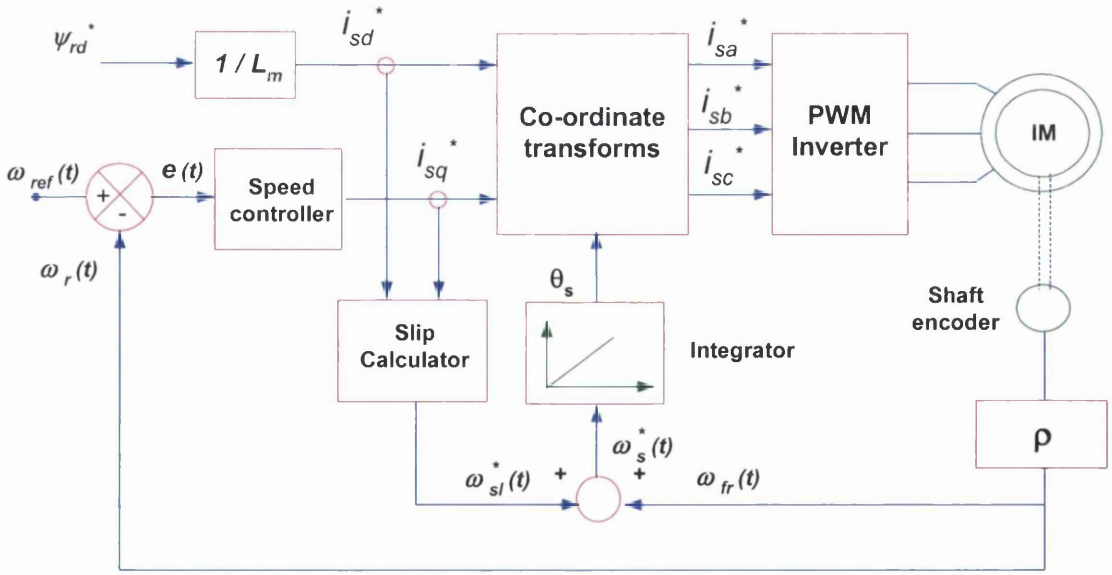


Figure 6-5: Configuration of an indirect field oriented induction motor drive with current-controlled PWM inverter.

It should be noted that the vector control configuration shown in Figure 6-5 employs a fast control of stator current with a current-controlled PWM inverter (Vas 1990; Murphy and Turnbull 1988). This arrangement is known as a current-fed rotor flux oriented system. The feedback three phase currents which are necessary to obtain the gate signals for the PWM inverter are not shown in Figure 6-5.

There exist another type of control, where the stator currents are indirectly controlled by appropriate variation of the stator terminal voltages. This can be implemented by using a typical voltage-controlled PWM inverter. This is known as a voltage-fed rotor flux oriented scheme. This scheme utilises the stator voltage equations in its implementation, and thus it is important to develop these equations for the voltage-fed case in field coordinates. Recall the stator voltage vector equation (3.50), derived in the synchronous reference frame,

$$\overline{v_s^f} = R_s \overline{i_s^f} + j\omega_s \overline{\psi_s^f} + \frac{d}{dt}\overline{\psi_s^f} \tag{6.20}$$

Also recall the stator flux vector equation (3.51), derived in the synchronous reference frame,

$$\overline{\psi_s^f} = L_s \overline{i_s^f} + L_m \overline{i_r^f} \tag{6.21}$$

Substitution of Equation (6.21) into (6.20) yields the following vector equation,

$$\overline{v_s^f} = R_s \overline{i_s^f} + j\omega_s (L_s \overline{i_s^f} + L_m \overline{i_r^f}) + \frac{d}{dt}(L_s \overline{i_s^f} + L_m \overline{i_r^f}) \tag{6.22}$$

The rotor current $\overline{i_r^f}$ can be replaced with use of Equation (6.9). Thus,

$$\overline{v_s^f} = R_s \overline{i_s^f} + j\omega_s (L_s \overline{i_s^f} + \frac{L_m}{L_r}\overline{\psi_r^f} - \frac{L_m^2}{L_r}\overline{i_s^f}) + \frac{d}{dt}(L_s \overline{i_s^f} + \frac{L_m}{L_r}\overline{\psi_r^f} - \frac{L_m^2}{L_r}\overline{i_s^f}) \tag{6.23}$$

Equation (6.23) can be rearranged to give,

$$\overline{v_s^f} = R_s \overline{i_s^f} + j\omega_s (L_s \overline{i_s^f} \sigma + \frac{L_m}{L_r} \overline{\psi_r^f}) + \frac{d}{dt}(L_s \overline{i_s^f} \sigma + \frac{L_m}{L_r} \overline{\psi_r^f}) \tag{6.24}$$

where σ is known as the total leakage factor of the induction motor and is defined as,

$$\sigma = 1 - \frac{L_m{}^2}{L_S L_r} \tag{6.25}$$

Substitution of Equation (3.47) into (6.24) and resolution of the vectors into their two axis components yields the following two real stator voltage equations,

$$v_{sd}^f = R_s i_{sd}^f - \omega_s L_s i_{sq}^f \sigma - \omega_s \frac{L_m}{L_r} \psi_{rq}^f + L_s \sigma \frac{di_{sd}^f}{dt} + \frac{L_m}{L_r} \frac{d\psi_{rd}^f}{dt} \tag{6.26}$$

$$v_{sq}^f = R_s i_{sq}^f + \omega_s L_s i_{sd}^f \sigma + \omega_s \frac{L_m}{L_r} \psi_{rd}^f + L_s \sigma \frac{di_{sq}^f}{dt} + \frac{L_m}{L_r} \frac{d\psi_{rq}^f}{dt} \tag{6.27}$$

As stated before, to force a field orientation to the rotor flux, the d-axis of the synchronous reference frame is aligned with the rotor flux, $\psi_{rq}^f = 0$ and $d\psi_{rq}^f / dt = 0$. By applying this constraint, the stator voltage equations (6.26) and (6.27) can be expressed in the rotor flux reference frame as follows,

$$v_{sd}^f = R_s i_{sd}^f - \omega_s L_s i_{sq}^f \sigma + L_s \sigma \frac{di_{sd}^f}{dt} + \frac{L_m}{L_r} \frac{d\psi_{rd}^f}{dt} \tag{6.28}$$

$$v_{sq}^f = R_s i_{sq}^f + \omega_s L_s i_{sd}^f \sigma + \omega_s \frac{L_m}{L_r} \psi_{rd}^f + L_s \sigma \frac{di_{sq}^f}{dt} \tag{6.29}$$

If a constant rotor flux operation is assumed, making ψ_{rd}^f = constant, then $d\psi_{rd}^f / dt = 0$. Thus equation (6.28) becomes,

$$v_{sd}^{f} = R_s i_{sd}^{f} - \omega_s L_s i_{sq}^{f} \sigma + L_s \sigma \frac{di_{sd}^{f}}{dt} \tag{6.30}$$

It can be seen from Equations (6.29) and (6.30) that there is an unwanted coupling in the stator voltage equations, where the q-axis stator current i_{sq}^{f} affects the d-axis of the stator voltage v_{sd}^{f}. Similarly the d-axis stator current i_{sd}^{f} and rotor flux ψ_{rd}^{f} affect the q-axis of the stator voltage v_{sq}^{f}.

Thus when implementing a voltage-fed field oriented control these coupling terms are undesirable, and it is necessary to introduce decoupling signals at the reference inputs in order to allow an independent control for the d and q stator current components and therefore avoid perturbing the rotor flux linkage as much as possible when responding to a change in load torque (Vas 1990). Figure 6-6 shows the block diagram of the voltage-fed indirect field oriented control for induction motor drive.

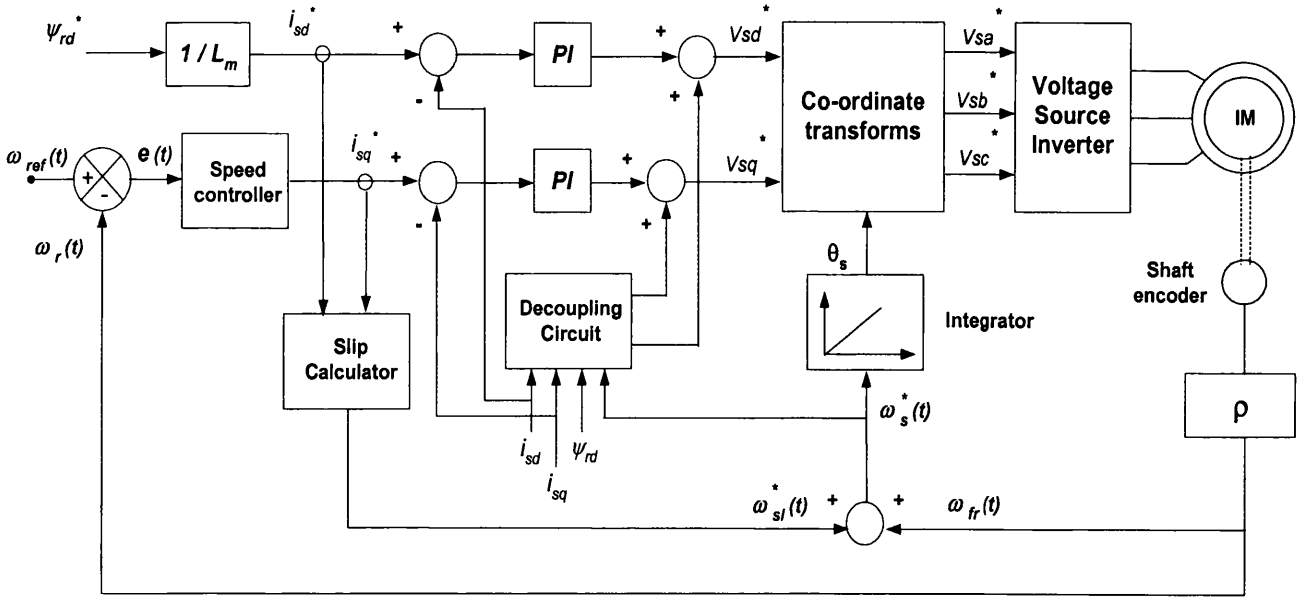


Figure 6-6: Configuration of voltage-fed indirect field oriented induction motor drive.

It can be seen from Figure 6-6 that in order to decouple the control of d and q stator currents i_{sd}^{f} and i_{sq}^{f} (rotor flux and torque producing components, respectively), the following decoupling components,

$$d_{sd} = -\omega_s L_s i_{sq}^f \sigma \tag{6.31}$$

$$d_{sq} = \omega_s L_s i_{sd}^f \sigma + \omega_s \frac{L_m}{L_r} \psi_{rd}^f \tag{6.32}$$

are added to the output of the two PI current controllers, which control i_{sd}^f and i_{sq}^f, respectively. The voltages on the output of these controllers are given by,

$$\hat{v}_{sd} = R_s i_{sd}^f + L_s \sigma \frac{di_{sd}^f}{dt} \tag{6.33}$$

$$\hat{v}_{sq} = R_s i_{sq}^f + L_s \sigma \frac{di_{sq}^f}{dt} \tag{6.34}$$

Therefore, adding Equation (6.31) to (6.33) yields the d-axis stator voltage component given in Equation (6.30). Similarly adding Equation (6.32) to (6.34) yields the q-axis stator voltage component given in Equation (6.29). Therefore, the decoupling circuit shown in Figure 6-6 contains the decoupling signals d_{sd} and d_{sq}, which are obtained from i_{sd}^f, i_{sq}^f, ψ_{rd}^f, and ω_s by using Equations (6.31) and (6.32), respectively.

In what follows, computer modelling and simulation of the *V/f* and indirect FOC schemes with the following speed control algorithms are presented: (1) fuzzy logic control (FLC), (2) variable structure control with sliding mode (VSC-SLM), and finally (3) the integrated fuzzy logic and sliding mode control (FSLMC).

In the *V/f* simulation model, a limiter has been introduced for the slip frequency command so that the motor operates below the rotor breakdown point. In the indirect FOC simulation model, voltage limiter has been introduced to limit the maximum phase voltage supplied to the motor model to 240V. Current limiter for the stator q-axis current is not used in the simulation.

6.4 Computer simulations of V/f and indirect FOC schemes using various control algorithms:

In the present work, computer simulations have been carried out in SIMULINK and MATLAB modelling environments to examine the characteristics of the induction motor drive dynamics using *V/f* and indirect vector control schemes, in which various speed control algorithms are employed.
This represents a very important step to initially design an efficient controller for the closed loop speed control of the induction motor, in which the design can be tested without costly and time-consuming experiments with hardware.

For a comprehensive transient study of the drive system, both electrical and mechanical transient dynamics must be considered in the simulation of the induction motor. For the *V/f* control scheme the induction motor model has been created using the dynamical equations, which are formulated in the synchronous reference frame, and rearranged into a state space representation as presented in Chapter 3. For the indirect vector control scheme the induction motor model based on the field coordinates, which was developed in this chapter is used. However, the decoupling circuit is omitted from the simulation. The stator and rotor current derivatives as well as the speed derivative of the dynamical model have been numerically integrated to compute their instantaneous values using Runge-Kutta algorithms.

The rated parameters of the simulated 3-hp induction motor have been determined experimentally and are given in Table 3.1. In order to bring out the fundamental operation of the proposed control algorithms in a simplified approach, the overall simulation model employs an ideal sinusoidal supply which allows perfect stator voltages and currents control.

6.4.1 SIMULINK representation of the V/f control system:

A simple SIMULINK scheme for *V/f* speed control of induction motor drive is presented in Figure 6-7. As can be seen, the scheme is comprised of three main parts grouped in separate SIMULINK blocks labelled as '*Speed controller*', '*Ideal sinusoidal supply*', and '*IM model*'. The first part represents the

speed control algorithms, which might be FLC, VSC-SLM, or FSLMC. The two input variables are the commanded reference speed (*W_ref*) and the feedback rotor speed (*W_r*) of the induction motor, while the output variable is the regulated slip frequency (*W_sl*). In addition, the digital implementation constraint to simulate the speed controller in a discrete form (digital system) is considered in this part. This can be achieved by utilising discrete-time functions with variable sampling rate T_s that defines different sampling steps. In this simulation, T_s is set to 500 μs.

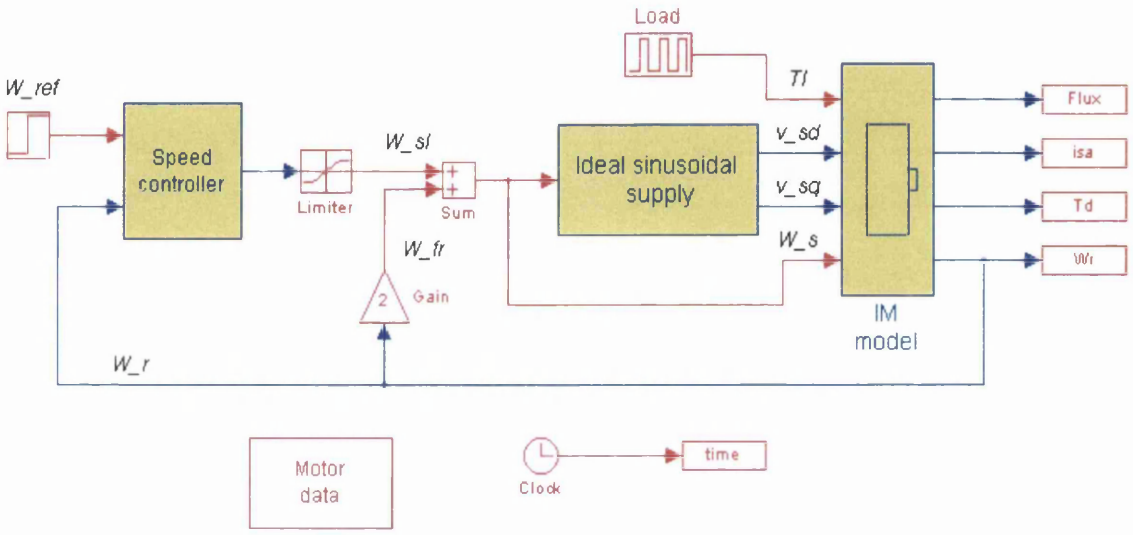


Figure 6-7: SIMULINK scheme of V/f speed control system.

The second part is an ideal sinusoidal three-phase source. It produces three-phase voltage waveforms in accordance with the stator frequency command (*W_s*) and the computed RMS voltage V_s. Phase and reference frame transformations are then used to express the generated phase voltages in synchronous reference frame in terms of their d and q axes (*v_sd*, *v_sq*). It should be noted that the field angle θ_s must be known in order to carry out the transformations. This angle can be computed by integrating the resulted stator frequency command with respect to time. The third part includes a set of differential equations that basically represent the transient dynamics of the induction motor. These equations, particularly the electrical ones, are modelled in a synchronous reference frame, which rotates at synchronous frequency defined by ω_s. The numerical solution of this model is obtained by selecting Runge-Kutta integration method from the **Simulation** menu in SIMULINK, in which the maximum integration step is set to be equal to T_s. The transient dynamics of some of the motor variables upon

direct start with fixed RMS voltage and constant frequency have been already presented in Chapter 3. In this simulation, the inputs to the IM model are the two stator voltage components (*v_sd*, *v_sq*) as well as the stator frequency (*W_s*) and load torque (Tl).

In consequents to the numerical integration performed in the IM model, the transient characteristics of the motor variables (stator and rotor currents, torque, speed, stator and rotor flux linkages, ...etc) are computed at each time interval defined by T_s. Each computed data is then stored into a corresponding MATLAB matrix (shown in Figure 6-7) as a column vector to be plotted against time for further dynamic analysis. In addition, the time vector is stored by feeding the Clock block into another MATLAB matrix denoted by *t* (also shown in Figure 6-7).

6.4.2 SIMULINK representation of the indirect FOC system:

A simple SIMULINK scheme for an indirect vector control of an induction motor drive system is exhibited in Figure 6-8. As can be seen, this scheme is comprised of similar parts to those employed in the *V/f* scheme configuration shown in Figure 6-7, with three additional subsystems.

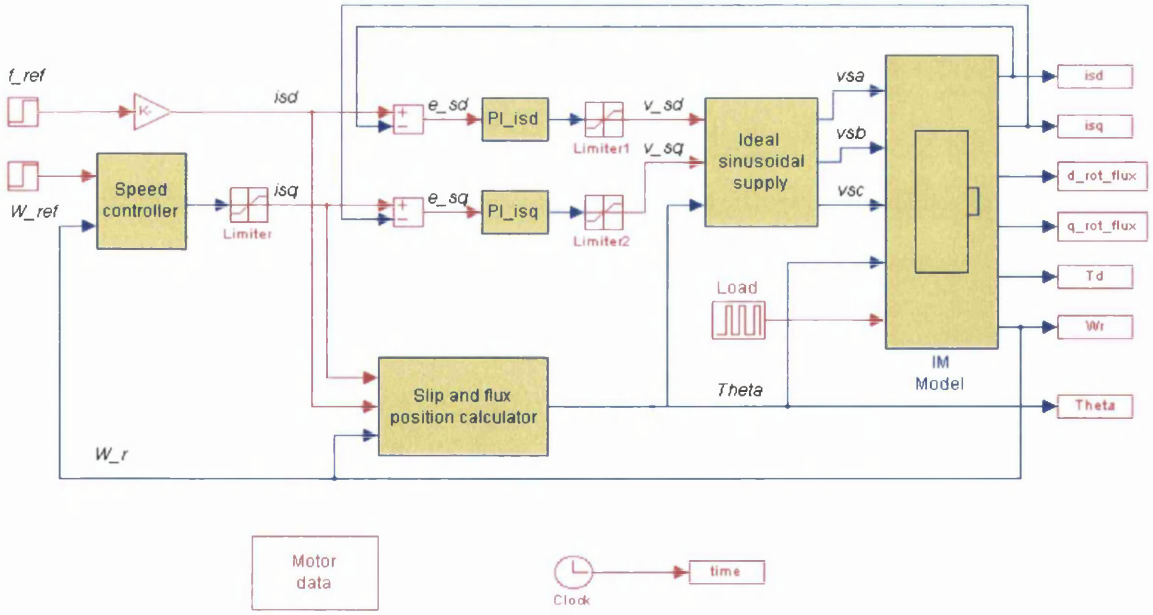


Figure 6-8: SIMULINK scheme of an indirect FOC system.

These subsystems are grouped in separate SIMULINK blocks labelled '*Slip and flux position calculator*', '*PI_isd*', and '*PI_isq*'. The rotor flux command is divided by the magnetising inductance L_m to obtain the flux current command i_{sd}^* in agreement with Equation (6.16). The commanded reference speed (*W_ref*) is compared with the feedback rotor speed (*W_r*) of the induction motor and the speed error serves as an input signal to the speed controller block. The output of the speed controller is the regulated torque current command i_{sq}^*. Next, both current commands i_{sd}^* and i_{sq}^* as well as the rotor speed are fed into the slip and flux position calculator block to compute the slip signal using equation (6.17).

In the same block, the computed slip signal is then added to the rotational speed frequency to obtain the stator frequency ω_s, which is then integrated to generate the field angle θ_s, in order to perform the transformations. There are two current control loops, one at each current axes, in which the stator currents are controlled in field co-ordinates using classical PI-controllers. Generally, it is useful to carry out these control loops in field co-ordinates because of the dc representation of the current signals. Thus, the PI-controllers (PI_isd, PI_isq) will effectively eliminate dc errors in the current signals, which will later coincide with the error elimination in the sinusoidal stator currents.

The inputs to the controllers are the two current error signals *e_sd* and *e_sq*, and the outputs are the demanded stator voltage components *v_sd* and *v_sq*. This type of control can be represented as,

$$\begin{bmatrix} v\_sd \\ v\_sq \end{bmatrix} = \left(K_P + \frac{K_I}{s} \right) \begin{bmatrix} e\_sd \\ e\_sq \end{bmatrix} \tag{6.35}$$

where K_P and K_I are the proportional and integral control gains of the PI-controller, respectively. The regulated voltage components (*v_sd*, *v_sq*) are then fed to the inverter circuit block in order to generate the three phase voltages that are used to drive the induction motor model. It should be noted that the decoupling circuit is omitted from the simulation.

6.4.3 Computer simulations of V/f and indirect FOC schemes with FLC:

In Chapter 4, a computer model for the FLC design has been developed in MATLAB programming environment, whereby it enables an easy design and redesign of the FLC algorithms. In the proposed control scheme, the actual inputs to the FLC system are the speed error e_o and its rate of change Δe_o, while the output is the slip frequency change $\Delta\omega_{sl}^*$ (*V/f* scheme) or the incremental torque current Δi_{sq}^* (indirect FOC scheme).

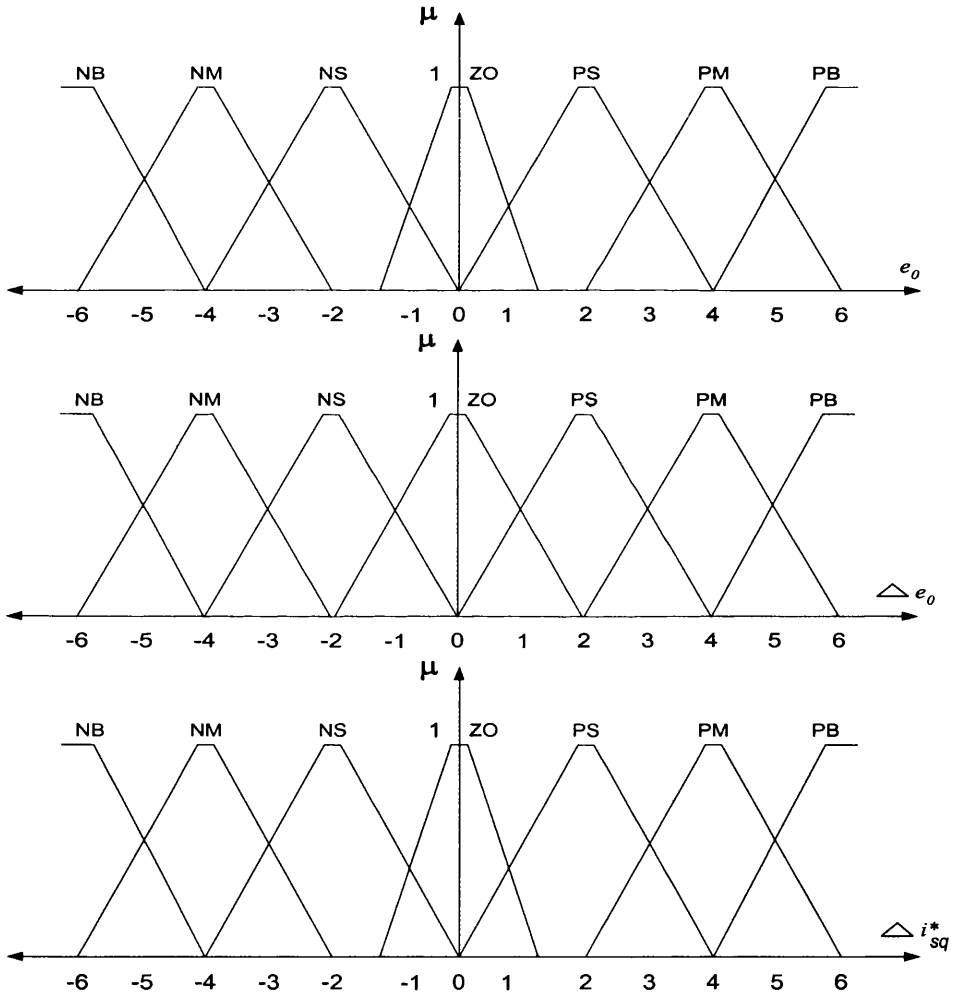


Figure 6-9: Membership functions representing the FLC variables in linguistics labels; (a) e_o membership functions; (b) Δe_o membership functions; (c) Δi_{sq}^* membership functions.

For a given speed error and its rate of change, the same control rules that have been developed and listed in Table 4.1 are used in this fuzzy control system. However, a new set of membership functions representing the FLC's input and output variables are re-constructed as shown in Figure 6-9, to meet the control requirements. The input variables e_o and Δe_o have been discretised into thirteen quantised levels, which are listed in Table 6-1.

Quantised level	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
e_o (r/min)	-170	-140	-113	-85	-57	-15	0	15	57	85	113	140	170
Δe_o (rad/sec^2)	-170	-140	-113	-85	-57	-29	0	29	57	85	113	140	170
Δi_{sq}^*	-100	-84	-66	-50	-33	-11	0	11	33	50	66	84	100

Table 6.1: The quantised levels of the error and change of error.

Δe_o \ e_o	*-6*	*-5*	*-4*	*-3*	*-2*	*-1*	*0*	*1*	*2*	*3*	*4*	*5*	*6*
-6	0	19	38	38	38	47	59	59	59	63	77	78	78
-5	-19	0	19	19	38	47	47	48	59	63	63	63	78
-4	-38	-19	0	19	38	38	38	48	59	59	59	63	78
-3	-38	-19	-19	0	19	19	38	47	47	47	59	63	63
-2	-38	-38	-38	-19	0	19	38	38	38	47	59	59	59
-1	-48	-48	-37	-19	-19	0	19	19	38	47	48	48	59
0	-59	-48	-37	-37	-37	-19	0	19	38	38	38	47	59
1	-59	-48	-48	-48	-37	-19	-19	0	19	19	38	47	47
2	-59	-59	-59	-48	-37	-37	-37	-19	0	19	38	38	38
3	-63	-63	-59	-48	-48	48	-37	-19	-19	0	19	19	38
4	-78	-63	-59	-59	-59	-48	-37	-37	-37	-19	0	19	38
5	-78	-63	-63	-63	-59	-48	-46	-46	-37	-19	-19	0	19
6	-78	-78	-78	-63	-59	-59	-59	-46	-37	-37	-37	-19	0

Table 6.2: The look-up decision table.

Consequently, the look-up decision table which is shown in Table 6.2 can be constructed from the fuzzy rules table (Table 4.1) and Table 6.1 using the simulated fuzzy system model as developed in Chapter 4. The output data are scaled up by a factor of 16.7. It should be noted that hunting and system instability around the set point (reference speed ω_{ref}) might result due to the coarse quantisation levels in the above decision table. Theoretically, their choice has a considerable effect on the FLC resolution. Thus large number of them makes the FLC system more sensitive to the observed variables and hence provides fine control output.

In practice, however, this increases the complexity of the control system and thus increases the computational time of the control loop in the microcontroller application. To tackle this problem, a novel solution has been proposed Liaw and Wang (1991), in which a simple integral controller is combined with the FLC system to minimise the steady state error and hence reduces the hunting effects. Figure 6-10 depicts the SIMULINK scheme for the proposed FLC system.

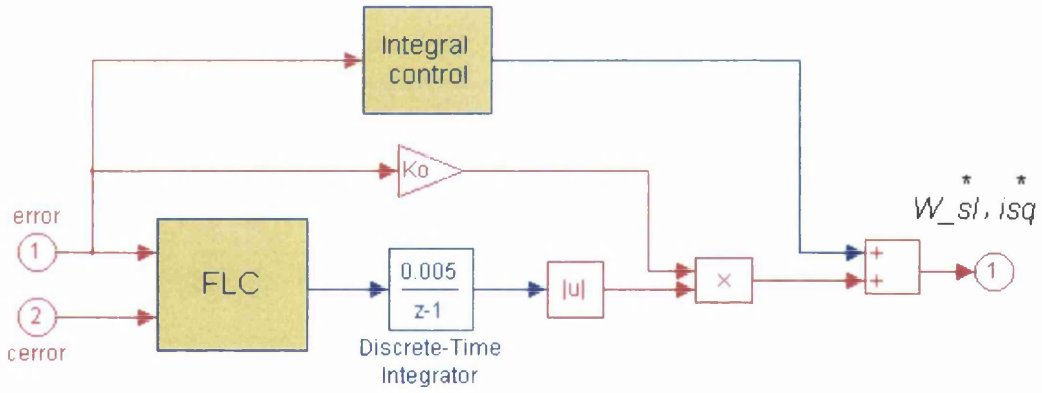


Figure 6-10: SIMULINK scheme of the proposed FLC system.

The FLC block embraces the designed fuzzy control algorithms in the form of a set of look-up decision statements for a given error and change of error as listed in Table 6.2. The effective output from the FLC block is basically the incremental slip frequency command $\Delta\omega_{sl}^{*}$ or the incremental torque current command Δi_{sq}^{*}. The addition of the gain *Ko* in the proposed control system has significantly improved the response of the rotor speed in the transient period.

6.4.3.1 Simulation results of V/f scheme with FLC:

The simulation results of the proposed FLC speed control system upon step change in reference speed using *V/f* scheme, under no load, from stand still to 1000 r/min are shown in Figure 6-11. The figure depicts the profiles of the rotor speed response, the stator current, and the electromagnetic torque, respectively.

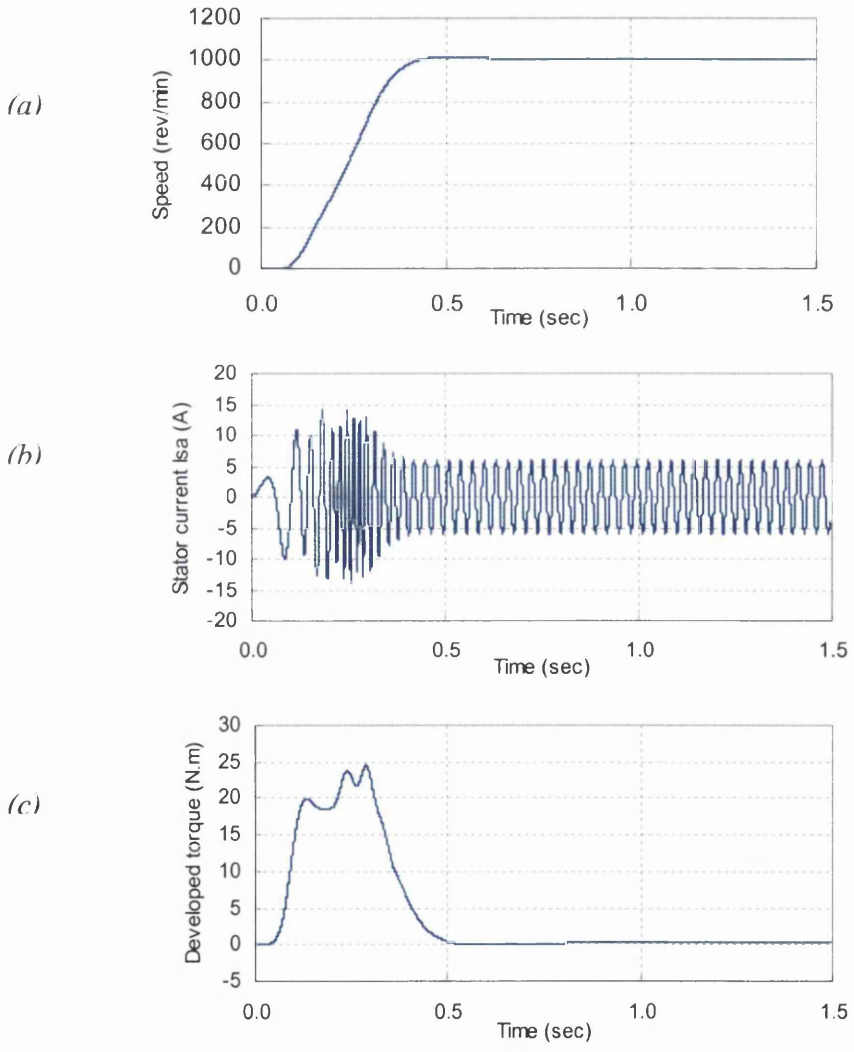


Figure 6-11: Simulation results of V/f scheme with the proposed FLC system, at no load condition, upon step change in reference speed; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque

As can be seen, the FLC control system initially has difficulty in driving the rotor shaft from stand still towards the reference speed because of the time needed to build up the flux in the stator and rotor circuits (Heber et al. 1997). Once the flux is established, the reference speed is then reached at reasonably fast rate with minute overshoot and no steady state error as expected. As the motor accelerates and the speed builds up, the electromagnetic torque develops and rises smoothly to a sufficient value dictated by the slip frequency command. The torque curve then converges to a steady state value dictated by the value of the friction constant.

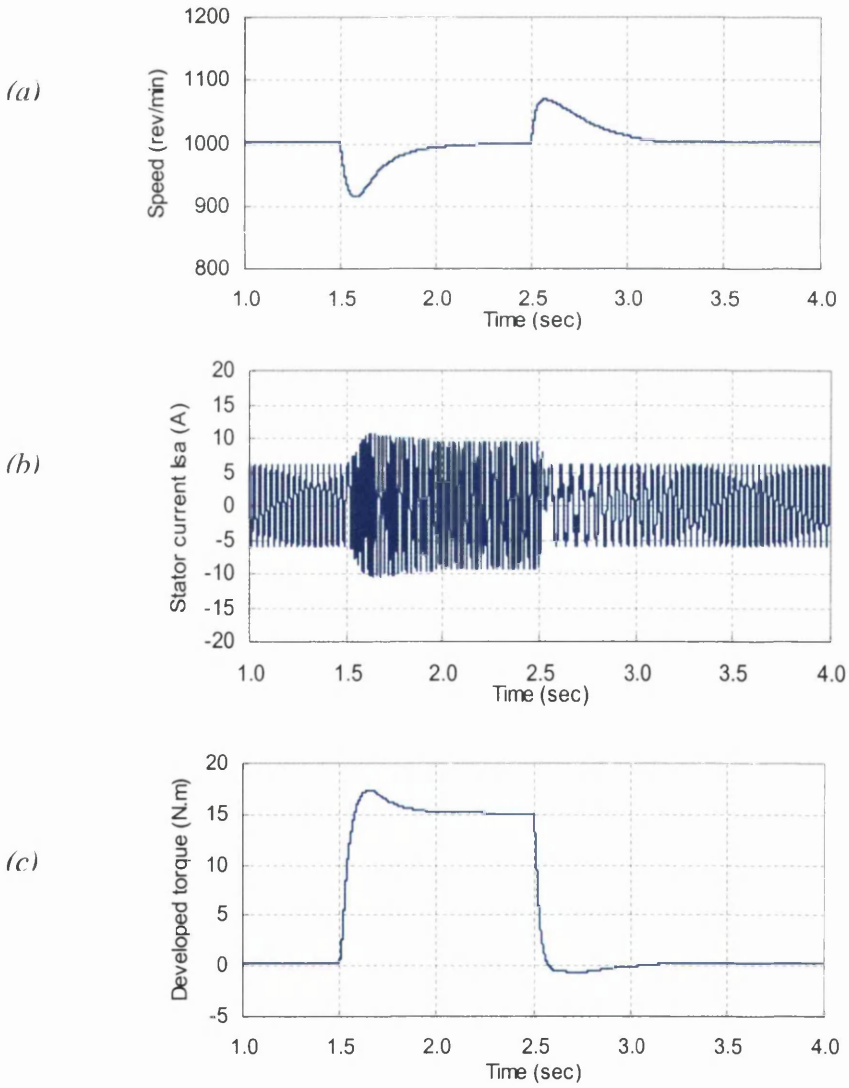


Figure 6-12: Simulation results of V/f scheme with the proposed FLC system during step change in load torque; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque;

The second simulation, Figure 6-12, examines the external load rejection capability of the FLC system when 100 % load was suddenly applied to the rotor shaft and later removed as the motor was initially running at 1000 r/min. In both cases, the FLC quickly brings back the rotor speed to the reference speed within a recovery time of 0.5 seconds with a maximum speed change of 80 r/min.

On the other hand, the performance of the FLC during parameter variations in the drive system is demonstrated in the final simulation test as shown in Figure 6-13. In this test, the rotor resistance is doubled while the motor shaft is still loaded with a 50 % of rated load. As expected, this causes a great degradation in the drive performance and as a result the rotor speed started to drop. Despite this degradation, the FLC system managed to restore the speed back to 1000 r/min within 1 second with a speed drop of 35 r/min.

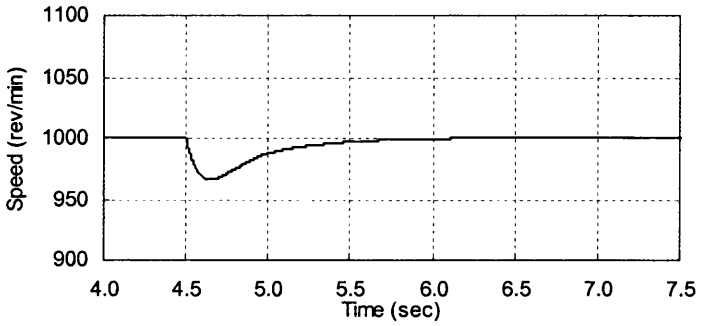
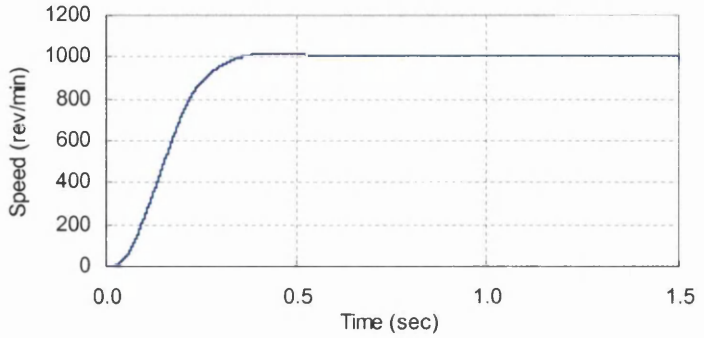


Figure 6-13: Speed profile of V/f scheme with the proposed FLC system during parameter variations; rotor resistance is doubled.

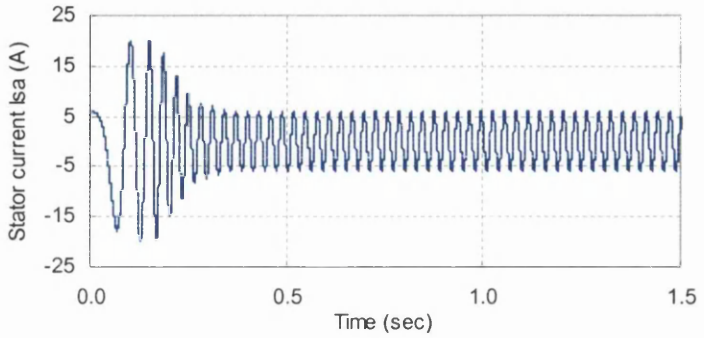
6.4.3.2 Simulation results of indirect FOC scheme with FLC:

The simulation tests carried out in here are the same as that performed in the previous section. Figure 6-14 illustrates the response profiles of some of the motor variables during step change in reference speed, at no load, from stand still to 1000 r/min. It should be noted that the quality of the stator current control is resolved by tuning the gains of the PI-controllers. Thus, proper selection of their values lead to a sufficient stator current control. In this test, the optimal values of the controller gains have been defined via continual simulation trials. These values are given as $K_P = 200$ and $K_I = 100$. It should be also noted that the decoupling circuit is omitted from the simulation.

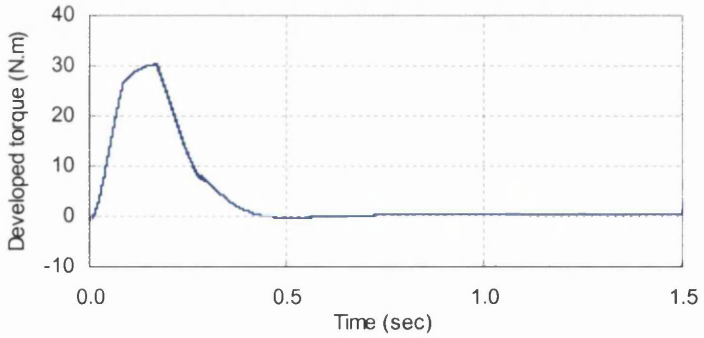
(a)



(b)



(c)



(d)

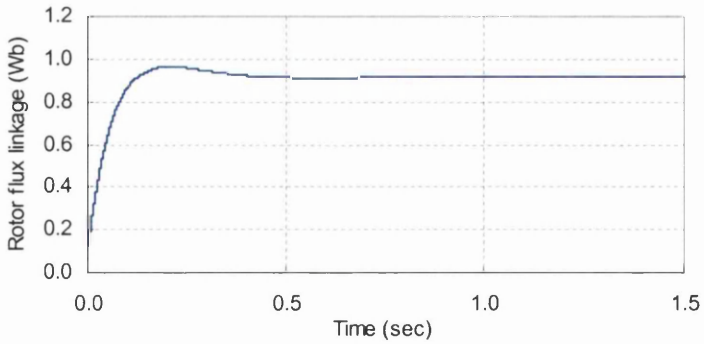


Figure 6-14: Simulation results of the indirect FOC scheme with the proposed FLC system, at no load condition, upon step change in reference speed; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque; (d) rotor flux linkage.

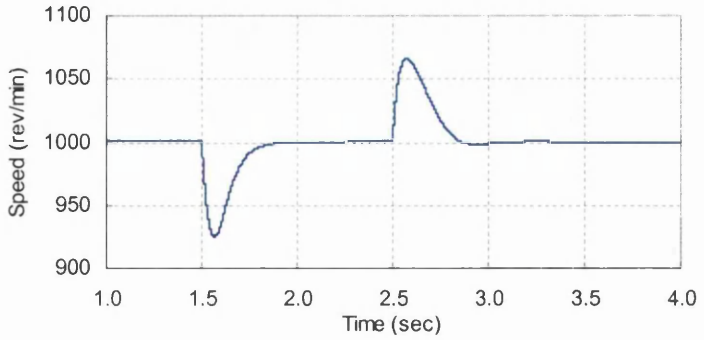
The rotor flux reference is set to be equal to 0.9-Wb. Inspection of this figure indicates that the rotor flux level rises exponentially to the reference value with small overshoot within a rise time of 0.2 second. This time is relatively equivalent to 3.3 rotor time constants (τ_r = 0.061 second). It can be seen that the rotor flux level remains approximately constant at the reference value throughout the simulation test. This figure also shows that the rotor speed of the drive system reached the reference value of 1000 r/min within almost 0.4 second, which is faster than that obtained with the *V/f* scheme in the previous section.

On the other hand, Figure 6-15 exhibits the dynamic responses of the rotor speed, the stator current, the developed torque and the rotor flux linkage due to step load changes when the motor was operated at 1000 r/min. At 1.5 second a 100 % load was applied to the rotor shaft and later removed at 2.5 second. In both cases the FLC system restored the rotor speed to the reference speed within 0.35 second with a maximum speed variation of 70 r/min. Moreover, it can be seen from the graph of the rotor flux linkage that due to the absence of the decoupling circuit, variation in the rotor flux during step loading and unloading can be observed.

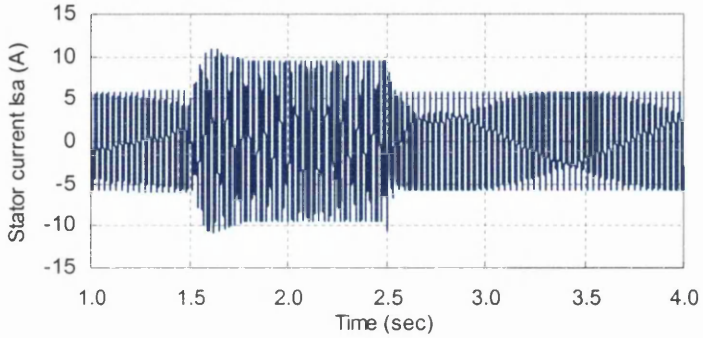
Finally, the performance of the FLC system during parameter variations in the drive system is demonstrated in the final simulation test as shown in Figure 6-16. In this test, the rotor resistance is doubled while the motor shaft is still loaded with a 50 % of rated load. When this happens, the rotor time constant starts to deviate from the correct value and thus affects the calculation accuracy of the slip frequency command. Consequently, this causes the rotor speed to vary and hence the performance of the speed drive system will be degraded. Despite this the FLC system managed to keep the rotor speed almost intact with insignificant speed variation as shown in Figure 6-16.

According to the results presented in this section, a considerable improvement in the transient dynamics of the induction motor can be obtained when the proposed FLC scheme with the indirect FOC are used to control the speed drive system. These improvements are perceived in the faster transient response of the motor speed during step change in reference speed and load torque in comparisons to those obtained with the scalar *V/f* system.

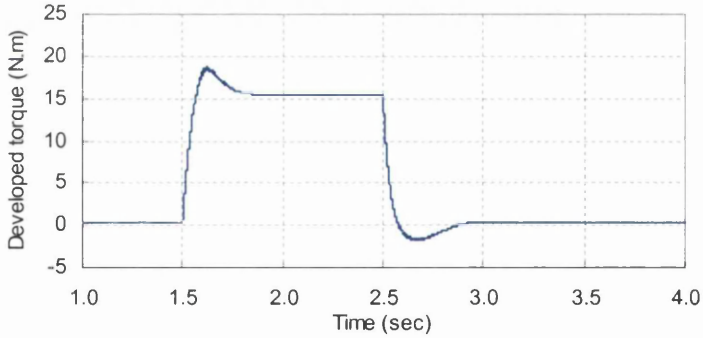
(a)



(b)



(c)



(d)

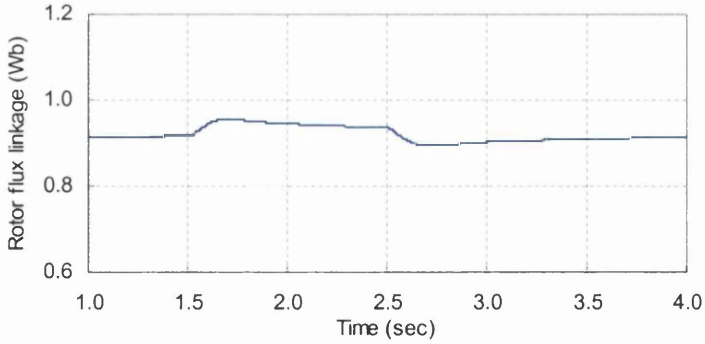


Figure 6-15: Simulation results of the indirect FOC scheme with the proposed FLC system, during step change in load torque; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque; (d) rotor flux linkage.

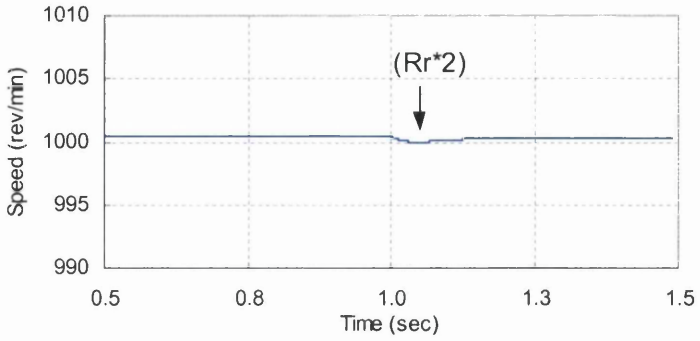


Figure 6-16: Speed profile of indirect FOC scheme with the proposed FLC system during parameter variations; rotor resistance is doubled.

6.4.4 Computer simulations of V/f and indirect FOC with the conventional VSC-SLM:

In the VSC-SLM control technique, the main objective is to force the state trajectory of the drive system to a pre-determined manifold surface, known as the sliding surface. The control laws are then designed so that the system trajectory always reaches the sliding surface $s(x)$ from any position in the system phase plane at finite time. Once on the sliding surface, the control structure is changed discontinuously to maintain the system trajectory sliding on the surface towards the origin, where the speed error e_o becomes zero. The basic concepts of designing a VSC-SLM control scheme for a speed drive system have been presented in Chapter 5, in which the system state variables are defined by the speed error and its rate of change. Figure 6-17 depicts the SIMULINK scheme for the conventional VSC-SLM control system.

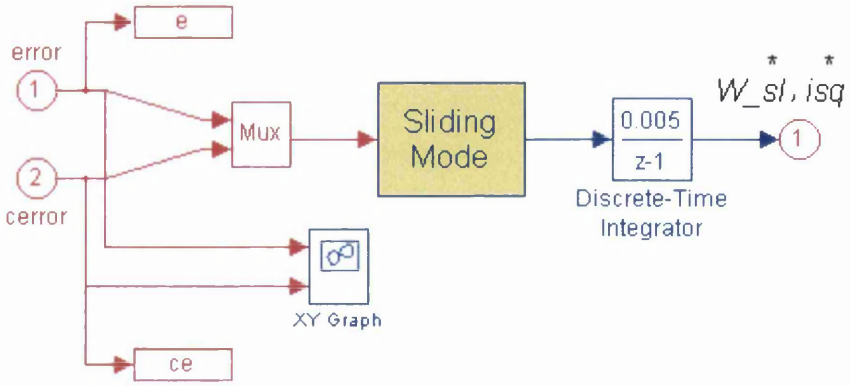


Figure 6-17: SIMULINK scheme of the conventional VSC-SLM system.

As can be seen, the above scheme configuration is comprised of one main control block, which is labelled as 'sliding mode'. This block specifies a user created function that has been written as an M-file, in which it embraces the designed control algorithms of the traditional VSC-SLM. Recalling the sliding mode control signal defined in Equation (5.9), which can be rewritten as

$$u_S = \varphi_1 x_1 + \varphi_2 x_2 \tag{6.36}$$

where the coefficients φ_1 and φ_2 are selected to ensure the system states move along the sliding line towards the origin point at which the error is zero. These gains are changed discontinuously such that,

$$\varphi_1 = \begin{cases} \alpha & \text{if } sx_1 > 0 \\ \beta & \text{if } sx_1 < 0 \end{cases}, \quad \varphi_2 = \begin{cases} \gamma & \text{if } sx_2 > 0 \\ \xi & \text{if } sx_2 < 0 \end{cases} \tag{6.37}$$

Nevertheless, a fast dynamic response without overshoot can be achieved by a good selection of a sliding surface that is defined by,

$$s(x) = Cx_1 + x_2, \qquad C > 0 \tag{6.38}$$

As pointed out in Chapter 5, according to the design configurations of the conventional VSC-SLM control systems, the invariance property is only present once the system states are on sliding surface, and thus robustness cannot be ensured before the sliding surface is reached. Therefore, the transient response is still sensitive to parametric variations and external disturbances. To overcome this problem, a simple solution has been proposed by Ho and Sen (1990), where two additional sliding line segments S_2 and S_3 are introduced as part of the sliding trajectories to provide robustness throughout the entire transient response (the sliding surface $s(x)$ is considered to be S_1). These segments are also known as the maximum acceleration segment and the maximum deceleration segment, respectively, which are defined as,

$$S_2 = x_2 - x_{2\_max} \tag{6.39}$$

$$S_3 = x_2 + x_{2\_max} \tag{6.40}$$

where x_2 and x_{2_max} are the acceleration state variable and the maximum limit of the motor acceleration, respectively. Consequently, the gain constants φ_1 and φ_2 ought to be derived in order to satisfy the reaching condition as given in Equation (5.7) for each sliding segment (S_1, S_2, and S_3). Using the inequality conditions outlined in Section 5.6, the values of the controller gains are defined and listed in Table 6.3. In what follows, the transient and steady state characteristics of some of the motor variables based on the conventional VSC-SLM control scheme are presented.

Controller gains	S_1	S_2	S_3
α	0.5	0.5	0.5
β	-0.5	-0.5	-0.5
γ	0.3	0.3	0.3
ξ	-0.3	-0.3	-0.3

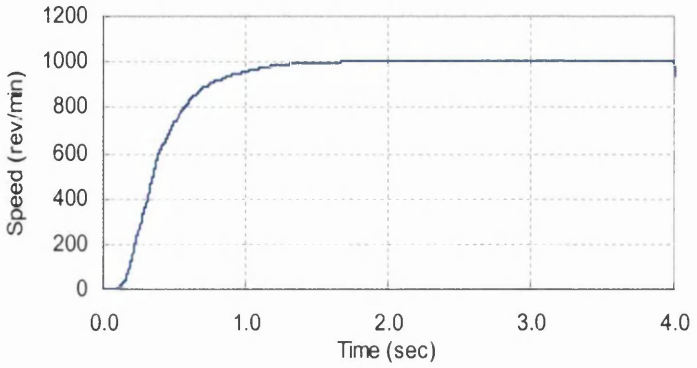
Table 6.3: Gain constants for the traditional VSC-SLM.

6.4.4.1 Simulation results of V/f scheme with the conventional VSC-SLM:

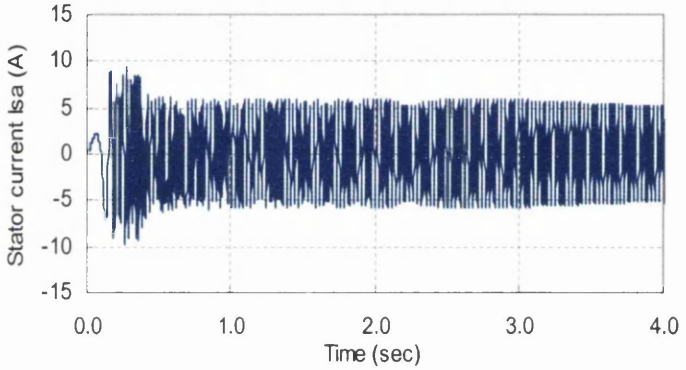
The simulation tests carried out in here are the same as that performed in the previous sections. The slope value C in Equation (6.9) has been selected to be 5 throughout the simulation tests, while x_{2_max} is set to 250 rad/sec^2. Figure 6-18 depicts the motor characteristics for a step change in reference speed, under no load, from stand still to 1000 r/min. Inspection of this figure shows that the reference speed is reached without overshoot and no steady state error as expected.

In contrast to the FLC performance, the speed response is slower due to the existence of the sliding line segments (S_1, S_2, and S_3), on which the system trajectory moves along to achieve the sliding mode operation. It should be noted that the speed response of the conventional VSC-SLM could be made faster if the slop value C of the sliding line is increased or by using larger controller gains. However, this might cause excessive overshoot in the speed response and might increase the chattering effects.

(a)



(b)



(c)

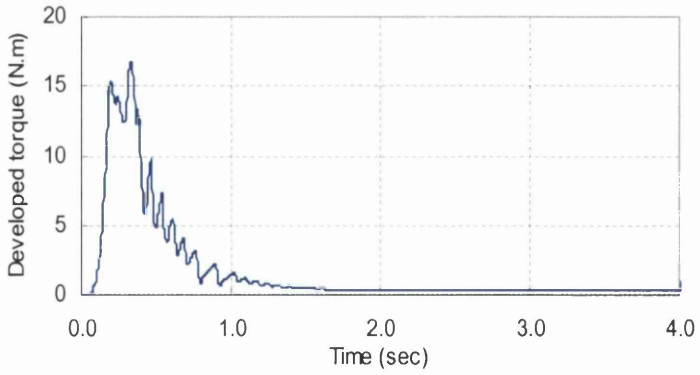


Figure 6-18: Simulation results of V/f scheme with the conventional VSC-SLM system during step change in reference speed; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque

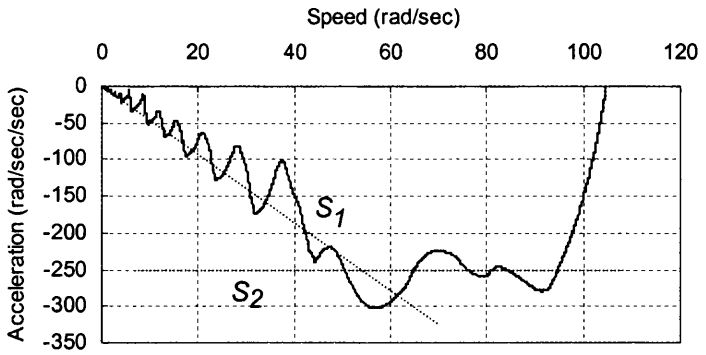


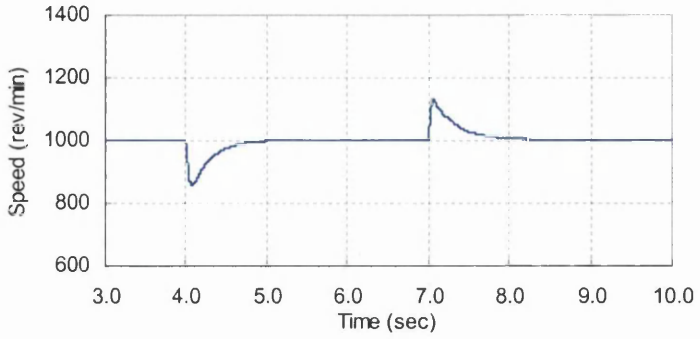
Figure 6-19: Phase plane trajectory of the conventional VSC-SLM during step change in the reference speed.

In addition, the system phase plane is depicted in Figure 6-19, in which the system trajectory follows the pre-specified sliding lines (S_1 and S_2) and thus satisfies the reaching condition of the VSC-SLM system. The second simulation, Figure 6-20, examines the performance of the VSC-SLM system during external load disturbances. Initially, the motor was running at a speed of 1000 r/min before a sudden 100 % load was applied to the rotor shaft at 4.0 second, and later removed at 7.0 second. In both cases, the VSC-SLM takes about 1 second to reinstate the speed to 1000 r/min with a maximum speed change of 120 r/min.

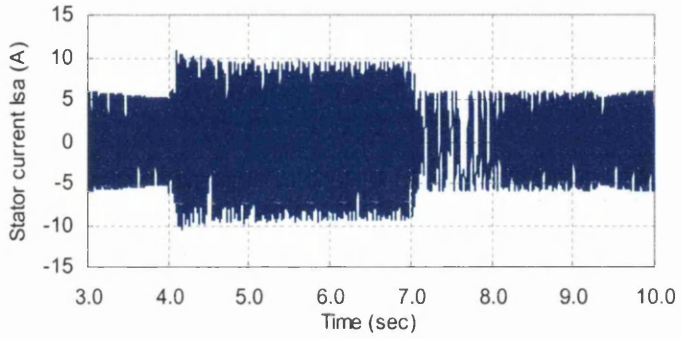
Next, the invariance property of the VSC-SLM is tested when the rotor resistance is doubled while the motor is still loaded with a 50 % of rated load. Despite the degradation in the drive performance, the VSC-SLM system managed to bring back the speed to 1000 r/min within 0.6 second with a speed drop of 30 r/min, as shown in Figure 6-21.

Inspection of the simulated results presented in Figure 6-18 and Figure 6-20 indicate that a good dynamical performance of the drive system can be obtained when the conventional VSC-SLM is used in the outer-loop speed control. However, in disparity to the FLC performance as shown in Figure 6-11 and Figure 6-12, the speed response is affected (slower response) by the inclusion of the sliding line segments (S_2, S_3) in order to secure robustness in the entire transient region.

(a)



(b)



(c)

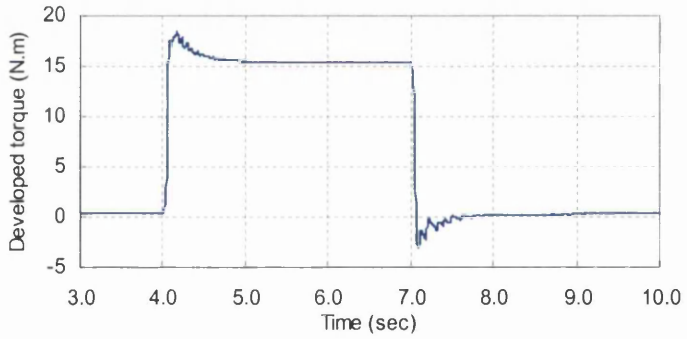


Figure 6-20: Simulation results of V/f scheme with the conventional VSC-SLM system during step change in load torque; (a) rotor speed response; (b) motor stator current; (c) developed torque;

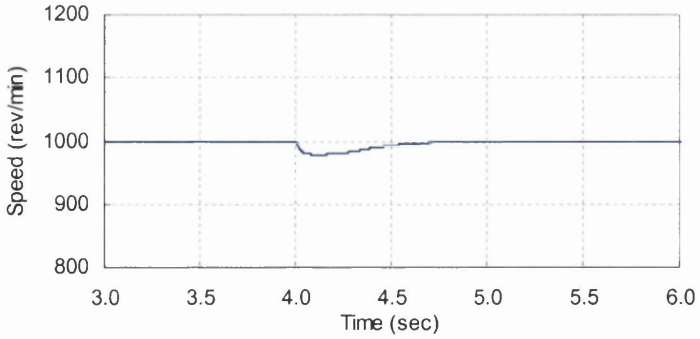


Figure 6-21: Speed profile of V/f scheme with the conventional VSC-SLM system during parameter variations; rotor resistance is doubled.

Generally, response improvement can be attained by using larger controller gains in the control signal, which is given in Equation (6.36). In this case, however, sever chattering effects might occur due to the excessive increase in the speed with which the state trajectory hits the sliding line. In contrary, Ho and Sen (1991) have proposed a novel method to provide a sufficient improvement in the transient dynamics of the speed drive system. This method implies adding the sliding line segment to the conventional control signal, Equation (6.36), to yield the following modified control signal,

$$u_m = \varphi_1 x_1 + \varphi_2 x_2 + K S_i, \qquad i = 1, 2, 3 \tag{6.41}$$

where K is a gain constant. It follows from the above equation that an additional control action is greatly contributed by the third term '$K S_i$' when the state trajectory is at a further distance from the sliding line. Once on the sliding line, however, the outcome of this term is relatively insignificant since the value of S_i becomes considerably small. Figures 6-22 and 6-23 show the response profile of the motor speed when the modified control signal, Equation (6.41), has been employed during a step change in reference speed and impact load disturbances, respectively. The gain K is set to be 0.6.

Inspection of Figure 6-22 indicates that enhanced speed acceleration can be accomplished with the modified VSC-SLM. However, insignificant improvement in load disturbance rejection can be observed as shown in Figure 6-23. Although the use of modified control signal can provide response improvement in the drive system, the chattering effects are slightly increased. These effects can be seen in the phase trajectory plane as shown in Figure 6-25.

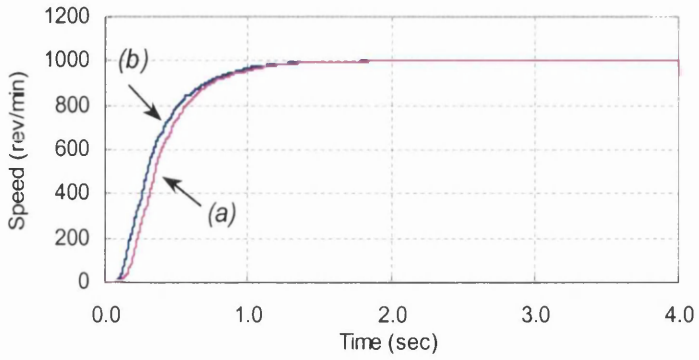


Figure 6-22: Speed response of V/f scheme during step change in reference speed with: (a) the conventional VSC-SLM system and (b) the modified VSC-SLM.

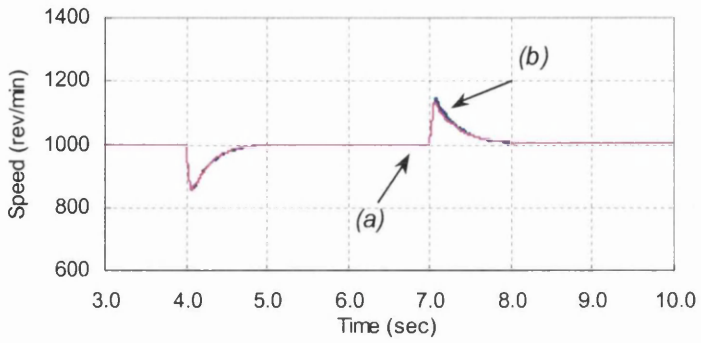


Figure 6-23: Speed response of V/f scheme during step change in load torque with: (a) the conventional VSC-SLM system and (b) the modified VSC-SLM.

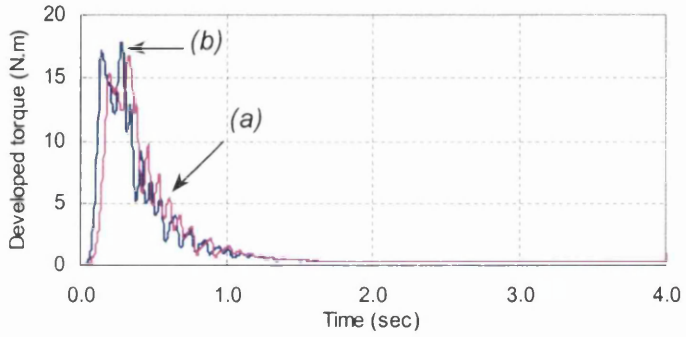


Figure 6-24: Electromagnetic torque profile of V/f scheme during step change in reference speed with: (a) the conventional VSC-SLM and (b) the modified VSC-SLM.

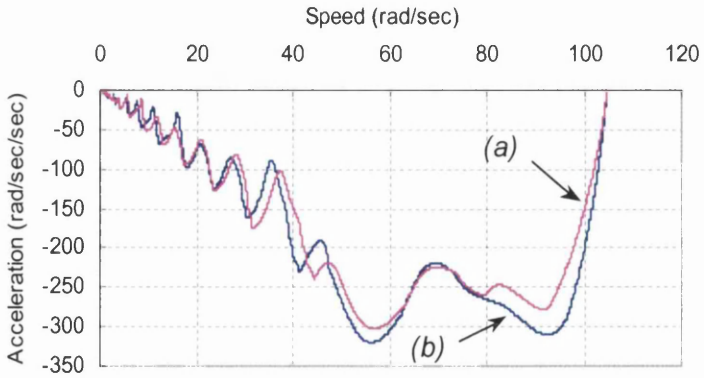


Figure 6-25: Phase plane trajectory of V/f scheme during step change in reference speed with (a) the conventional VSC-SLM and (b) the modified VSC-SLM.

6.4.4.2 Simulation results of indirect FOC scheme with the conventional VSC-SLM:

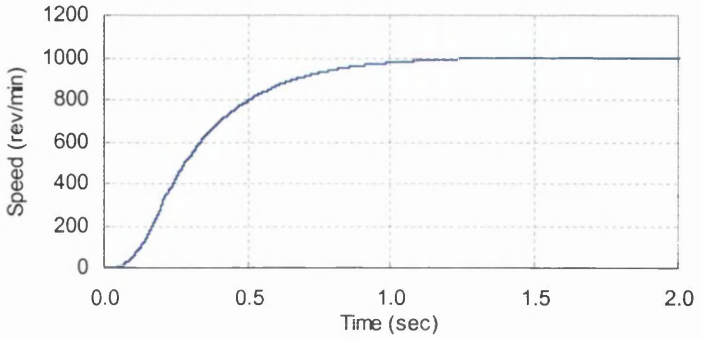
The simulation tests carried out in here are the same as that performed in the foregoing sections. Figure 6-26 presents the transient characteristics of the speed drive system during step change in reference speed, at light running, from standstill to 1000 r/min. The values of the PI-controllers gains are kept the same as before, $K_P = 200$ and $K_I = 100$. Moreover, the rotor flux reference is set to be equal to 0.9-Wb.

Inspection of these results indicates that the rotor flux level ramps up exponentially to the reference value with no overshoot within a rise time of 0.25 second, which is nearly equivalent to 4 rotor time constants ($\tau_r = 0.061$ second). This figure also shows that the rotor speed of the drive system reached the reference value of 1000 r/min within almost 1.3 second, which is faster than that obtained with the *V/f* scheme in the previous section. In contrast to the *V/f* scheme, the indirect field oriented control scheme provides better current control strategy that is accountable to produce smooth sinusoidal stator current signals.

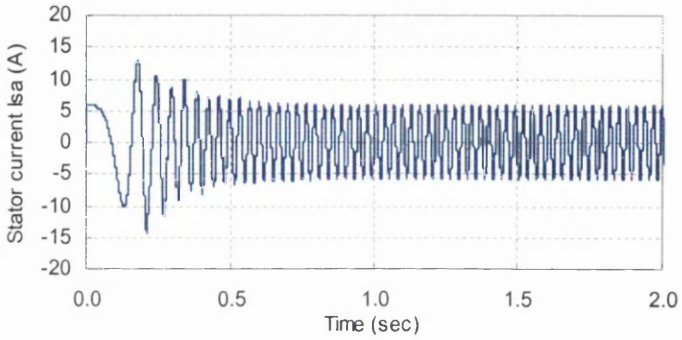
On the other side, Figure 6-27 exhibits the transient characteristics of the speed drive system during step load changes when the motor was initially operated at 1000 r/min. At 2.5 second, a 100 % load was applied to the rotor shaft and later removed at 4.0 second. In both cases the conventional VSC-SLM restores the rotor speed to the reference speed within 1 second with a maximum speed variation of 70 r/min. Moreover, it can be seen from the graph of the rotor flux linkage that due to the absence of the decoupling circuit, variation in the rotor flux during step loading and unloading can be observed.

Finally, the performance of the VSC-SLM system during parameter variations in the drive system is demonstrated in the final simulation test as shown in Figure 6-28. In this test, the rotor resistance is doubled while the motor shaft is still loaded with a 50 % of rated load. Consequently, this causes the rotor speed to drop. Despite this drop, the conventional VSC-SLM system manages to keep the rotor speed nearly constant with a rather small speed drop of 2 r/min.

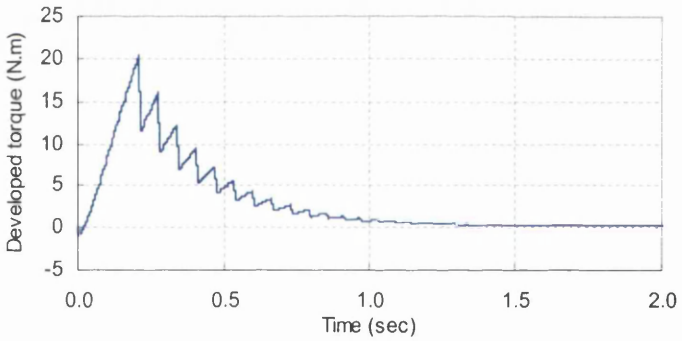
(a)



(b)



(c)



(d)

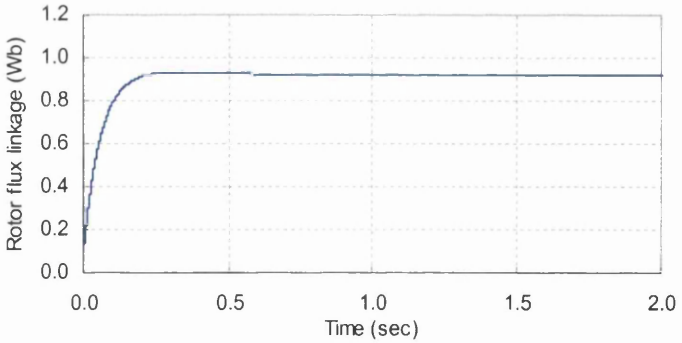
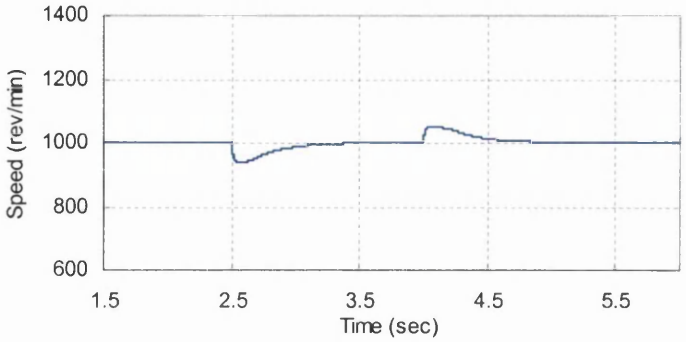
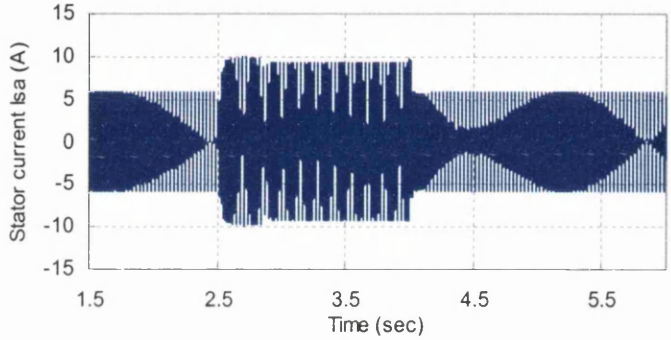


Figure 6-26: Simulation results of indirect FOC scheme with the conventional VSC-SLM system during step change in reference speed; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque; (d) rotor flux linkage.

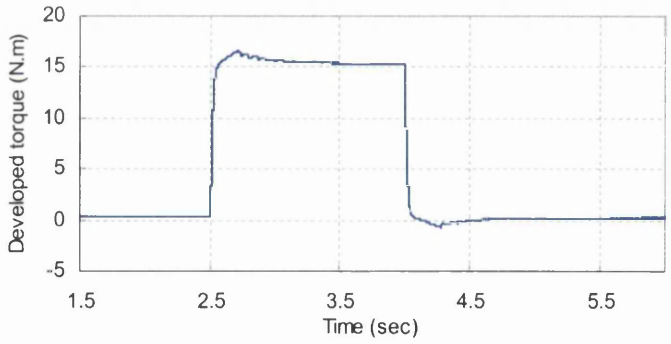
(a)



(b)



(c)



(d)

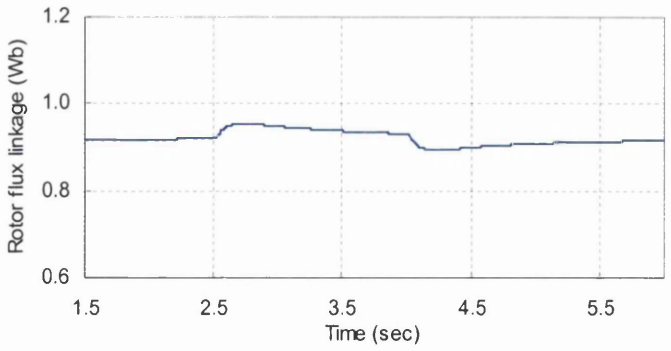


Figure 6-27: Simulation results of indirect FOC scheme with the conventional VSC-SLM system during step change in load torque; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque; (d) rotor flux linkage.

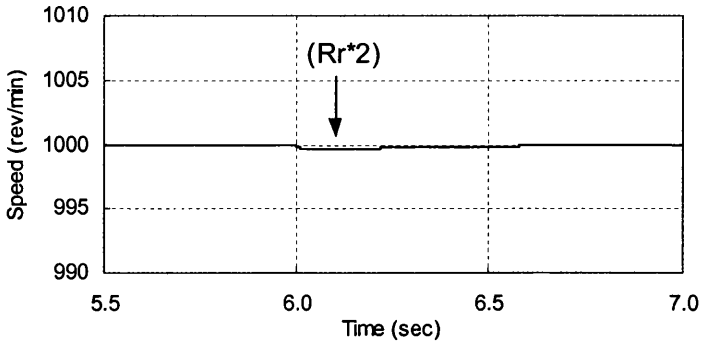


Figure 6-28: Speed profile of indirect FOC scheme with the conventional VSC-SLM system during parameter variations; rotor resistance is doubled.

While the research efforts persist in the improvement of VSC-SLM systems, some researchers have suggested the integration of the FLC algorithms in the VSC-SLM design. This led to the development of a new method that combines both control algorithms in order to reduce the chattering effects inherited in VSC-SLM and improve the transient dynamics of the speed drive system.

6.4.5 Computer simulations of V/f and indirect FOC schemes with the Proposed FSLMC:

It is known that the transient dynamics of VSC-SLM systems consist of two parts that characterise two modes of operation. The first part is known as the '*reaching mode*' (or non-sliding mode), while the second part is known as the '*sliding mode*'. The distinctive features of VSC-SLM are its robustness and invariance to system parameter perturbations and external load disturbances. In the conventional VSC-SLM control systems, however, the invariance property only exists once the system states are in the sliding mode, and thus the system is still sensitive to parametric variations in the reaching mode.

Therefore, an alternative design of VSC-SLM system has been proposed, in Chapter 5, with which the desired dynamics in the reaching mode as well as in the sliding mode are achieved and hence robustness is retained. The essence of the new design resides in the introduction of the fuzzy logic algorithms in the VSC-SLM system design, which allows faster hitting (small reaching time) and

smooth transition between reaching and sliding modes. The design procedures of the new control algorithm, FSLMC, for a speed drive system have been presented in Chapter 5, in which the system state variables are defined by the speed error and its rate of change. Once more, the SIMULINK scheme for the FSLMC control system is depicted in Figure 6-29.

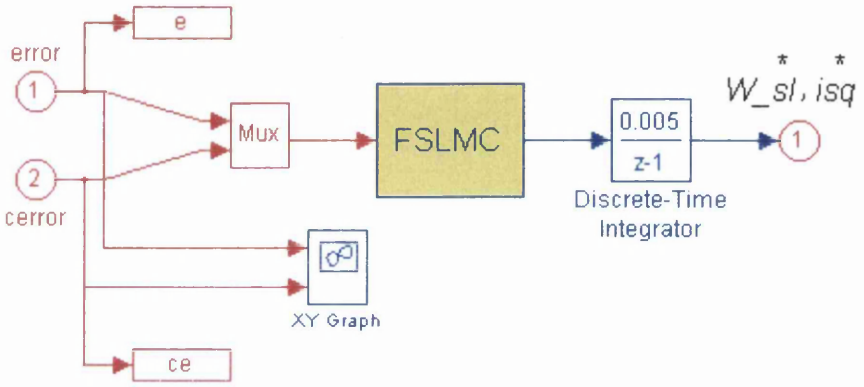


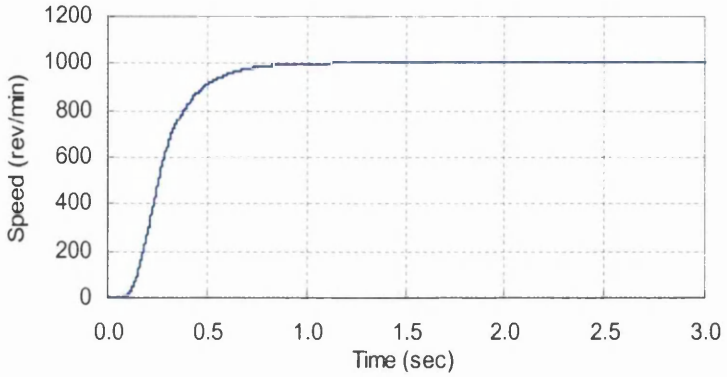
Figure 6-29: SIMULINK scheme of the proposed FSLMC system.

Again, the control algorithms of the proposed FSLMC have been specified in a user created function that has been written using MATALB commands and stored as an M-file. In contrast to the conventional VSC-SLM, the additional sliding line segments (S_2, S_3) are no longer required in this control scheme to provide robustness in the entire transient stage. This is simply because of the inherent dynamic structure of the control signal in the reaching mode, in which the invariant property of the control system is retained. In what follows, the transient and steady state characteristics of some of the motor variables based on the proposed FSLMC scheme are presented.

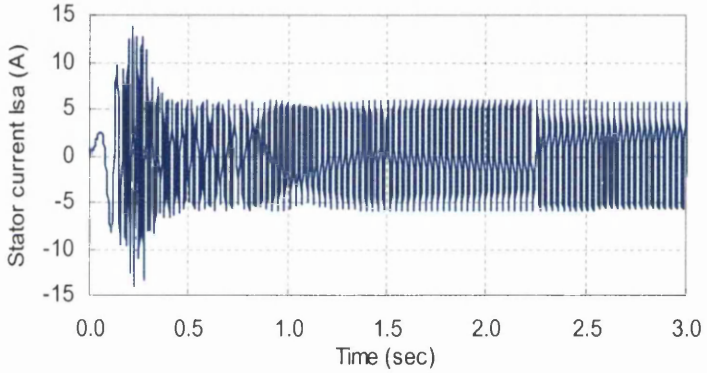
6.4.5.1 Simulation results of V/f scheme with the proposed FSLMC:

The simulation tests carried out in here are the same as that performed in the previous sections. The values of the controller gains α, β, γ, and ξ are 0.32, -0.32, 0.07, -0.07, respectively. The slope value C of the sliding line has been selected to be 7 throughout the simulation tests. Figures 6-30 and 6-31 depict the motor characteristics for a step change in reference speed and load torque, respectively.

(a)



(b)



(c)

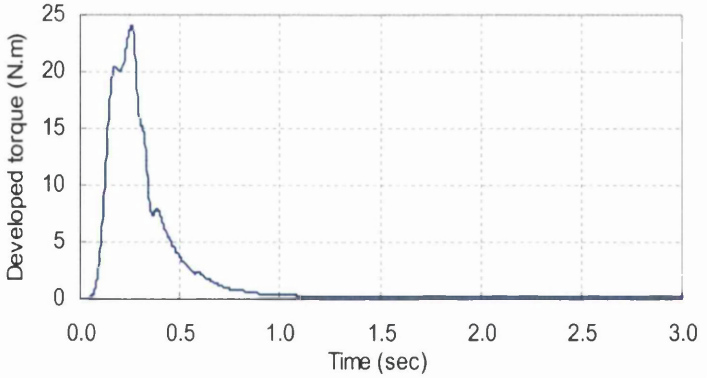
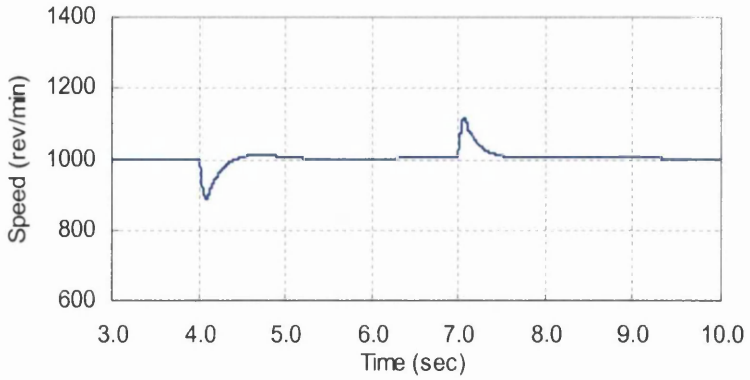
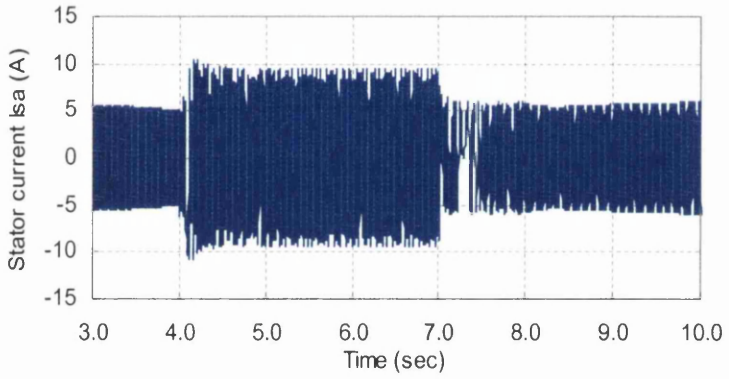


Figure 6-30: Simulation results of V/f scheme with the proposed FSLMC system during step change in reference speed; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque;

(a)



(b)



(c)

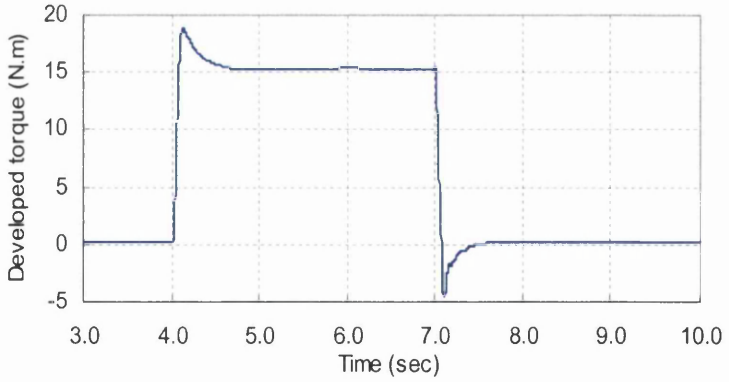


Figure 6-31: Simulation results of V/f scheme with the proposed FSLMC system during step change in load torque; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque.

Inspection of these results indicates that tremendous improvements in the transient characteristics of the drive system can be obtained when the proposed FSLMC is employed in the outer-loop speed control instead of the conventional VSC-SLM. These improvements are conspicuous in the faster transient response of the motor speed and the significant reduction of chattering effects, which are the main causes of high frequency ripples in the torque signal. Likewise, Figure 6-32 indicates a significant reduction of the chattering effects in the state trajectory of the drive system during step change in reference speed.

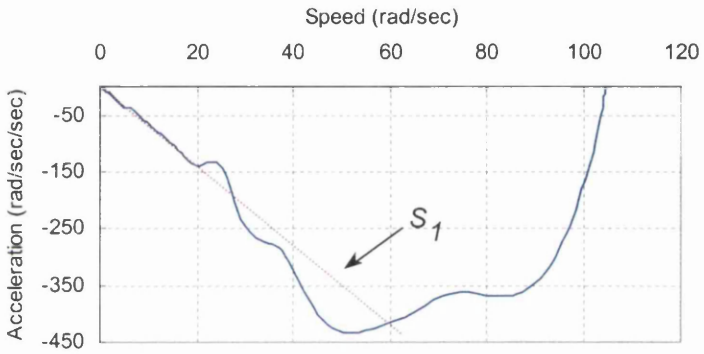


Figure 6-32: Phase plane trajectory of the proposed FSLMC during step change in reference speed.

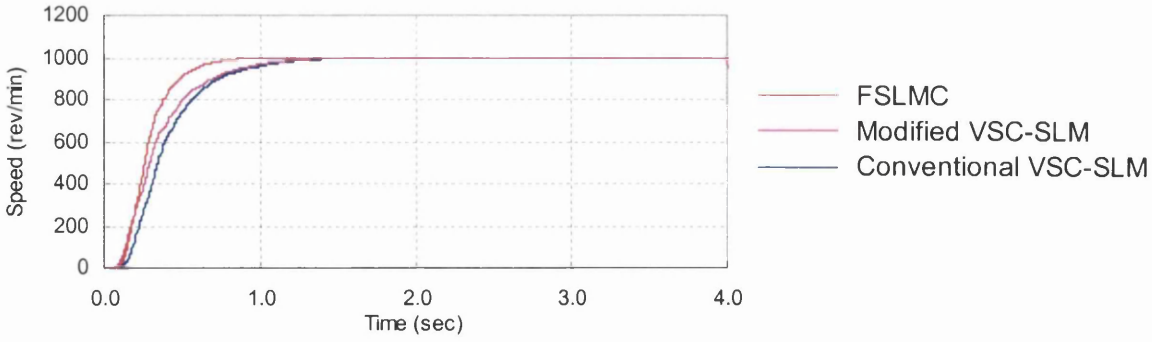


Figure 6-33: Speed responses of V/f scheme during step change in reference speed with the conventional VSC-SLM system, the modified VSC-SLM and the proposed FSLMC.

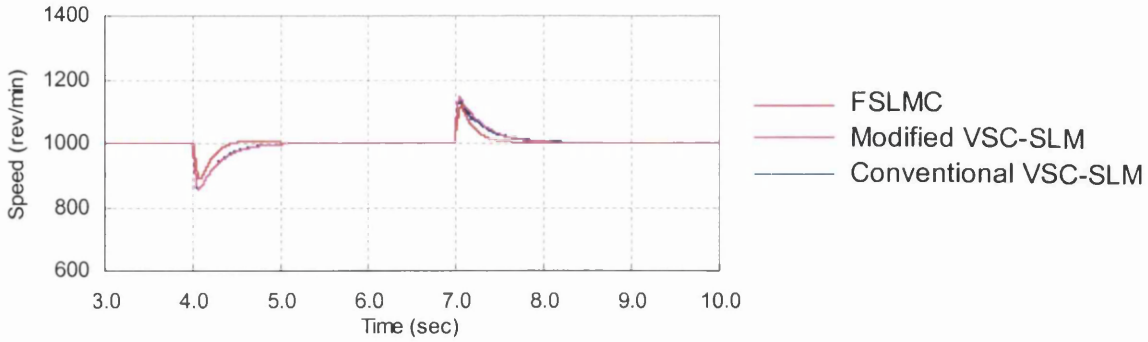


Figure 6-34: Speed responses of V/f scheme during step change in load torque with the conventional VSC-SLM system, the modified VSC-SLM and the proposed FSLMC.

To further highlight the dynamic improvements of the drive system when using the new controller with *V/f* control scheme, Figures 6-33 and 6-34 show three traces of speed response that correspond to the conventional VSC-SLM, modified VSC-SLM and the proposed FSLMC, during step change in reference speed and load torque, respectively. These figures clearly show the dynamics superiority of the FSLMC controller over both the conventional and the modified VSC-SLM controllers.

On the hand, to complete the dynamics comparisons, the speed response of the induction motor with a simple fuzzy logic controller is added to Figures 6-33 and 6-34. The results are shown in Figures 6-35 and 6-36 during step change in reference speed and load torque, respectively. These figures show the superiority of fuzzy logic controller (FLC) over the other three controllers with a very simple structure.

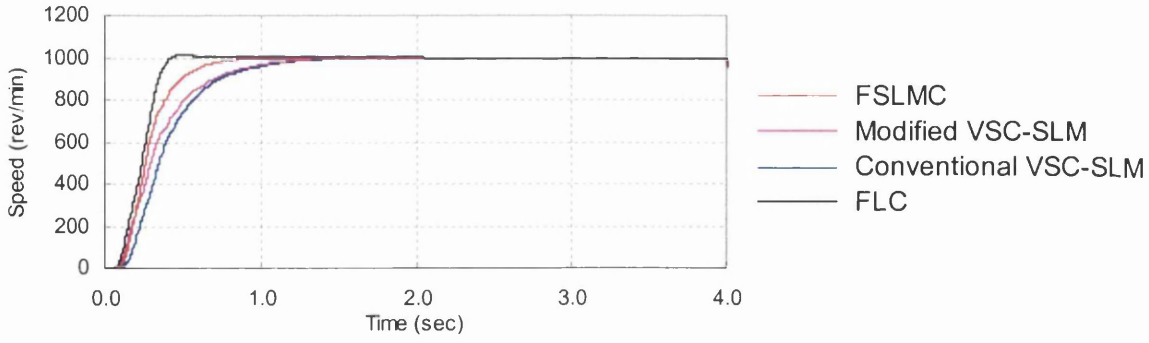


Figure 6-35: Speed responses of V/f scheme during step change in reference speed with the conventional VSC-SLM system, the modified VSC-SLM, the FSLMC, and the FLC.

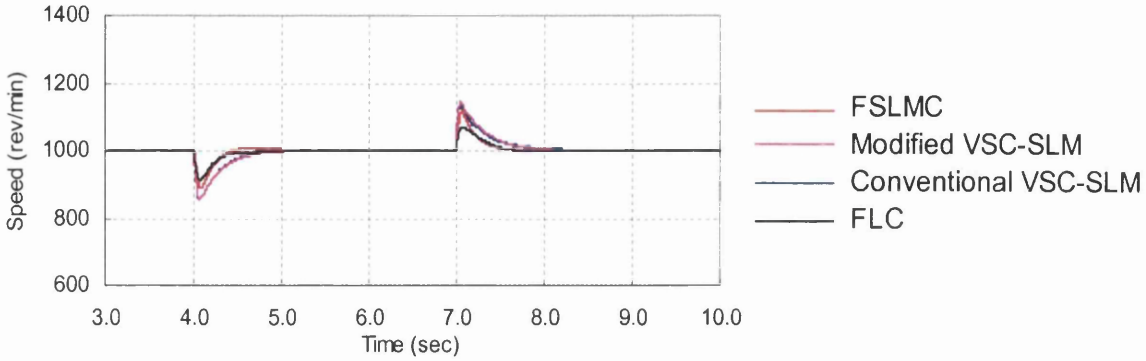


Figure 6-36: Speed responses of V/f scheme during step change in load torque with the conventional VSC-SLM system, the modified VSC-SLM, the FSLMC, and the FLC.

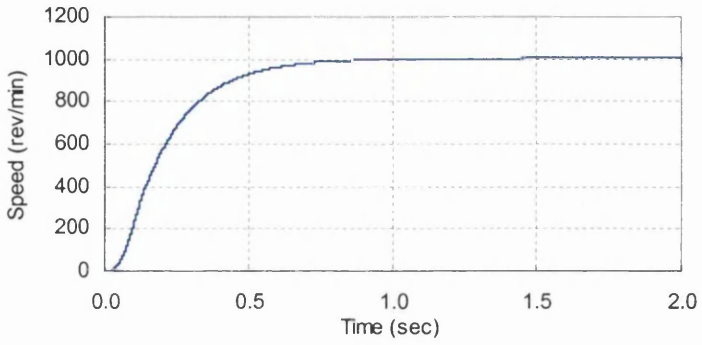
6.4.5.2 Simulation results of indirect FOC scheme with the proposed FSLMC:

The simulation tests carried out in here are the same as those performed in the foregoing sections. The transient characteristics of the speed drive system during step change in reference speed, at light running, from standstill to 1000 r/min are demonstrated in Figure 6-37. The values of the PI-controllers gains are kept the same as before, K_P = 200 and K_I = 100. In addition, the rotor flux reference was kept at 0.9-Wb.

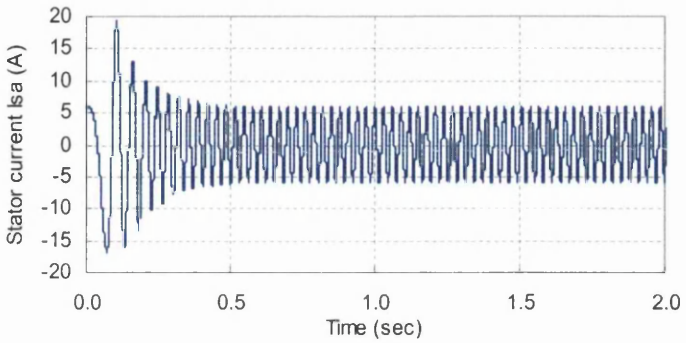
Inspection of these results shows that the rotor speed of the drive system reached the reference value of 1000 r/min within almost 0.9 second, which is faster than that obtained with the *V/f* scheme in the previous section.

Figure 6-38 exhibits the transient characteristics of the speed drive system during step load changes when the motor was initially operated at 1000 r/min. At 2.0 second, a 100 % load was applied to the rotor shaft and later removed at 3.5 second. In both cases the FSLMC system brings back the rotor speed to the reference speed within 0.5 second with a maximum speed change of 50 r/min. Moreover, it can be seen from the graph of the rotor flux linkage that due to the absence of the decoupling circuit, variation in the rotor flux during step loading and unloading can be observed

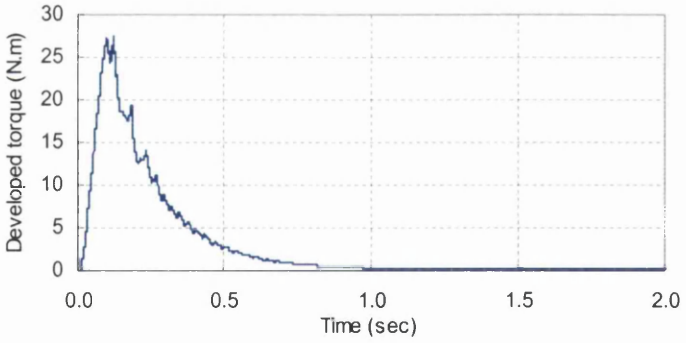
(a)



(b)



(c)



(d)

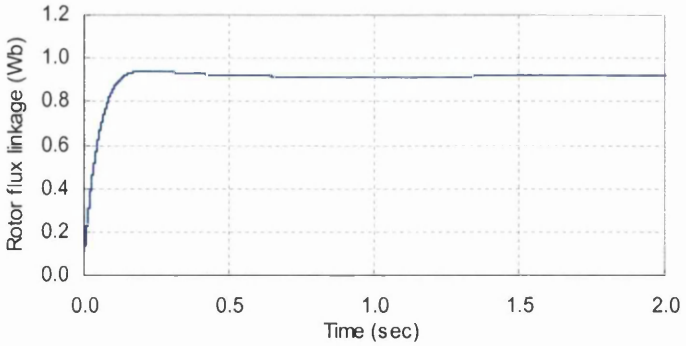
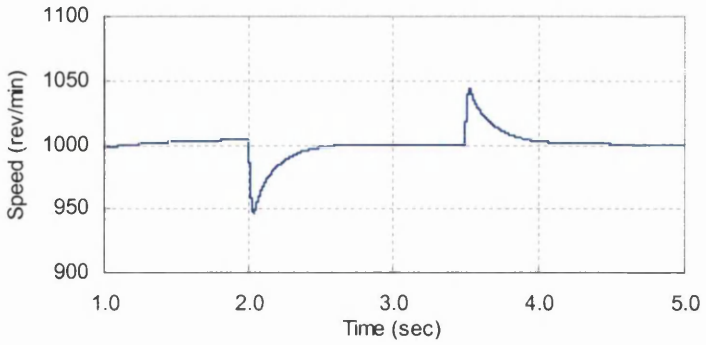
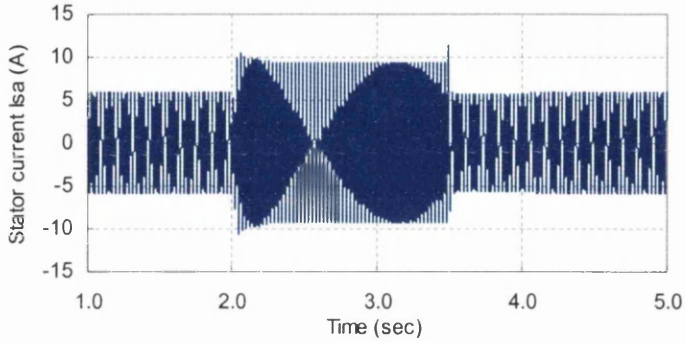


Figure 6-37: Simulation results of indirect FOC scheme with the proposed FSLMC system during step change in reference speed; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque; (d) rotor flux linkage.

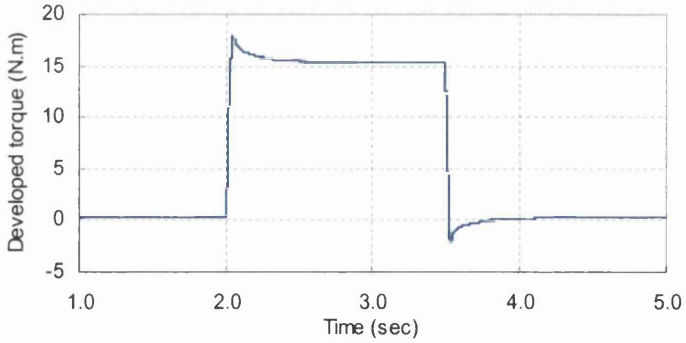
(a)



(b)



(c)



(d)

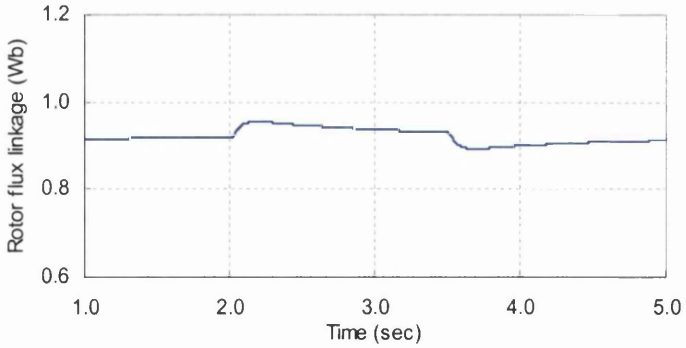


Figure 6-38: Simulation results of indirect FOC scheme with the proposed FSLMC system during step change in load torque; (a) rotor speed response; (b) motor stator current; (c) electromagnetic torque; (d) rotor flux linkage.

6.5 Summary:

In this chapter, the general control concepts regarding scalar and vector controls were briefly reviewed. This is followed by a short discussion on field oriented control techniques with the aid of the transient analysis of the induction motor, which was presented in Chapter 3. A performance comparison between various control algorithms for speed regulation of induction motor was assessed via computer modelling and simulation. The control algorithms are fuzzy logic control (FLC), variable structure control with sliding mode (VSC-SLM) and the proposed control scheme that introduces fuzzy logic concepts into the design of the sliding mode algorithms (FSLMC). The following are the major points given in this chapter:

1. In the scalar control methods for induction motors, the motor model is considered just for a precise steady state operation to provide a satisfactory steady state performance. Therefore, using this type of control strategy, the best dynamic performance of the induction motor can not be achieved during the transient stage. This is due to the highly non-linear coupled characteristics of the induction motor.

2. Vector control has emerged as an alternative control system, which is used to de-couple the control of the stator current components: the flux-producing current component i_{sd} and the torque-producing current component i_{sq}. Thus, an independent control of flux and torque can be achieved, which is similar to the control principles of a dc machine. Such control method builds upon the good steady state performance obtained with the scalar control, and can give excellent dynamic characteristics in the transient period.

3. Inspection of the obtained simulation results shows tremendous improvements in the transient characteristics of the drive system when the proposed FSLMC was employed in the outer-loop speed control in comparison to those obtained with the conventional VSC-SLM. These improvements are conspicuous in the faster transient response of the motor speed and the extreme reduction of chattering effects, which are the main causes of high frequency ripples in the torque signal. However, the results also showed the superiority of fuzzy logic controller (FLC) over the conventional VSC-SLM and the FSLMC controllers with a very simple structure.

Chapter 7, Digital implementation of the proposed speed drive system:

7.1 Introduction:

As deduced from chapter 6, a fast transient response insensitive to parameters and operating condition changes can be achieved using a simple FLC controller. It has been shown through simulation studies that the FLC scheme can attain a good transient performance for the speed drive system in comparison to the conventional SLMC and the improved FSLMC. Moreover, the design simplicity of the FLC system made it virtually attractive for the ease of practical implementation of the drive system. The digital implementation of the FLC algorithms is utilised on a single chip, Intel 80C196KC 16-bit embedded microcontroller, a low cost derivative of the MCS-96 architecture. This chip is chosen because it is cheap, reliable and capable of providing a wide range of computational tasks. Full description of this microcontroller is given in Appendix B. Furthermore, all the programming was done in assembly language using the ASM196 compiler.

In this chapter, general description of the experimental set-up for the FLC speed drive system is exhibited. This is followed by a detailed presentation of the software development for the closed loop speed control operation. This also includes the generation of the three-phase sinusoidal PWM waveforms and the techniques of digital speed measurements. The design methods for a digital filter based on the rolling average concept to improve the speed measurement are also highlighted. In addition, flowcharts are presented to demonstrate the operation of the developed software and full list of the program code is given in Appendix E.

7.2 Experimental set-up of the speed drive system:

The experimental set-up of the complete speed drive system is shown in Figure 7-1. As can be seen, the drive system is comprised of five primary sections. The first section is the full-wave uncontrolled diode bridge that converts three-phase ac voltage to a rectified dc voltage. Since the uncontrolled bridge has no firing delays, the drive system will operate at a constant power factor.

The second section is the DC-link (or DC bus), which embraces an energy storing capacitor, connected across the DC-link terminals to provide a smooth and constant dc voltage. The value of the capacitor bank that is used in this drive system is 6600 μF with a rated voltage of 350 V. The third section includes the PWM inverter bridge and the gate driver circuits. With the inverter bridge, the dc input voltage is converted back to a symmetrical ac output voltage of desired magnitude and frequency. A variable voltage magnitude can be obtained by varying the gain of the inverter circuit whereas the frequency is determined by the rate at which the power devices are switched on and off. The drive unit is also equipped with a dynamic breaking circuit. However, it should be noted that this circuit would not be used in this work.

The power devices used in the inverter bridge are the insulated gate bipolar transistors (IGBTs), which have very high switching capability. The IRGPH40F IGBT type is used in here, which can withstand a maximum voltage of 1200 V, a maximum current of 58 A and a continuous current of 29 A. To allow freewheeling to take part in the inverter operation, ultra fast recovery diodes are connected across the IGBT power devices. On the other hand, the gate driver circuits are designed using the push-pull configuration (Abed 1995). Furthermore, a so-called reciprocal circuit is included so that the switching signals of any two power devices on the same arm of the inverter bridge are complemented. A dead time is also included between these complementary signals to prevent short circuiting the DC-link. In addition, the TLP759 opto-couplers are used to isolate the microcontroller board from the driver circuits. Protection circuits against over-current, over-voltage or short circuit are also included.

The Sinusoidal Pulse Width Modulation (SPWM) technique with a switching frequency of 3.5 kHz is used to toggle the power devices on and off, in order to develop near-sinusoidal current waveforms with minimum low-order harmonic contents. Although the chosen value of the switching frequency is small, it can offer good torque characteristics when compared to older control scheme such as the six-step inverter drive. It should be noted that higher switching frequencies can reduce torque pulsation, which is normally very notable at low speed operation and hence acoustic noise from the motor is minimised. However, the switching losses become excessive because of the increase in the switching frequency, and this can heave the temperature of the power devices, and thus shortening their lives. Therefore, a trade-off has to be made between better control performance and system efficiency.

The forth section is the microcontroller system portion, in which certain control tasks are performed as summarised in Figure 7-2. The Intel EV80C196KC microcontroller single evaluation board is used in this drive system. The board is mainly delivered with an 80C196KC single chip microprocessor, two 16-bit and one 8-bit memory banks, UART for host communication, and digital I/O facilities. Detailed description of the microcontroller architecture is given in Appendix B. In brief, the microprocessor chip has the following features:

a) oscillator frequency of 16 MHz, with a minimum instruction execution time of 0.375 μS,
b) over 100 instructions,
c) 64K-byte addressable memory space with six basic addressing modes,
d) binary and decimal arithmetic operations,
e) register to register architecture,
f) 256 bytes of on-chip RAM, which consists of register file and Special Function Register (SFR),
g) the register file can be accessed as bytes (8 bits), words (16 bits), or double words (32 bits).
h) flexible interrupt facility,
i) 8-channel A/D converter with 8 or 10 bit successive approximations,
j) two 16-bit Timers, which can be clocked by the internal clock generator or external sources.

Finally, the fifth section includes the three-phase ac induction motor, dc shunt generator and a speed sensor. The induction motor used in this investigation is a star-connected 3-hp, 415V, 1415 r/min, 4 poles motor. The load torque is produced by connecting a variable resistor bank to the output terminals of the DC generator, which is mechanically coupled to the shaft of the induction motor. The rated values of the DC generator are as follows: 110 V, 1.5 kW, 1440 r/min, and 13.6 A.

The speed of the motor is sensed via an optical shaft encoder, which is mounted on the induction motor shaft. This encoder is capable of generating 1024 pulses per one rotor revolution. The accuracy of the speed measurement can be improved by increasing the number of pulses per revolution and hence reducing the speed detection period. This can be achieved by feeding the output of the encoder phases A and B (which lag each other by 90°) to a simple logic circuit in order to double the number of pulses per revolution.

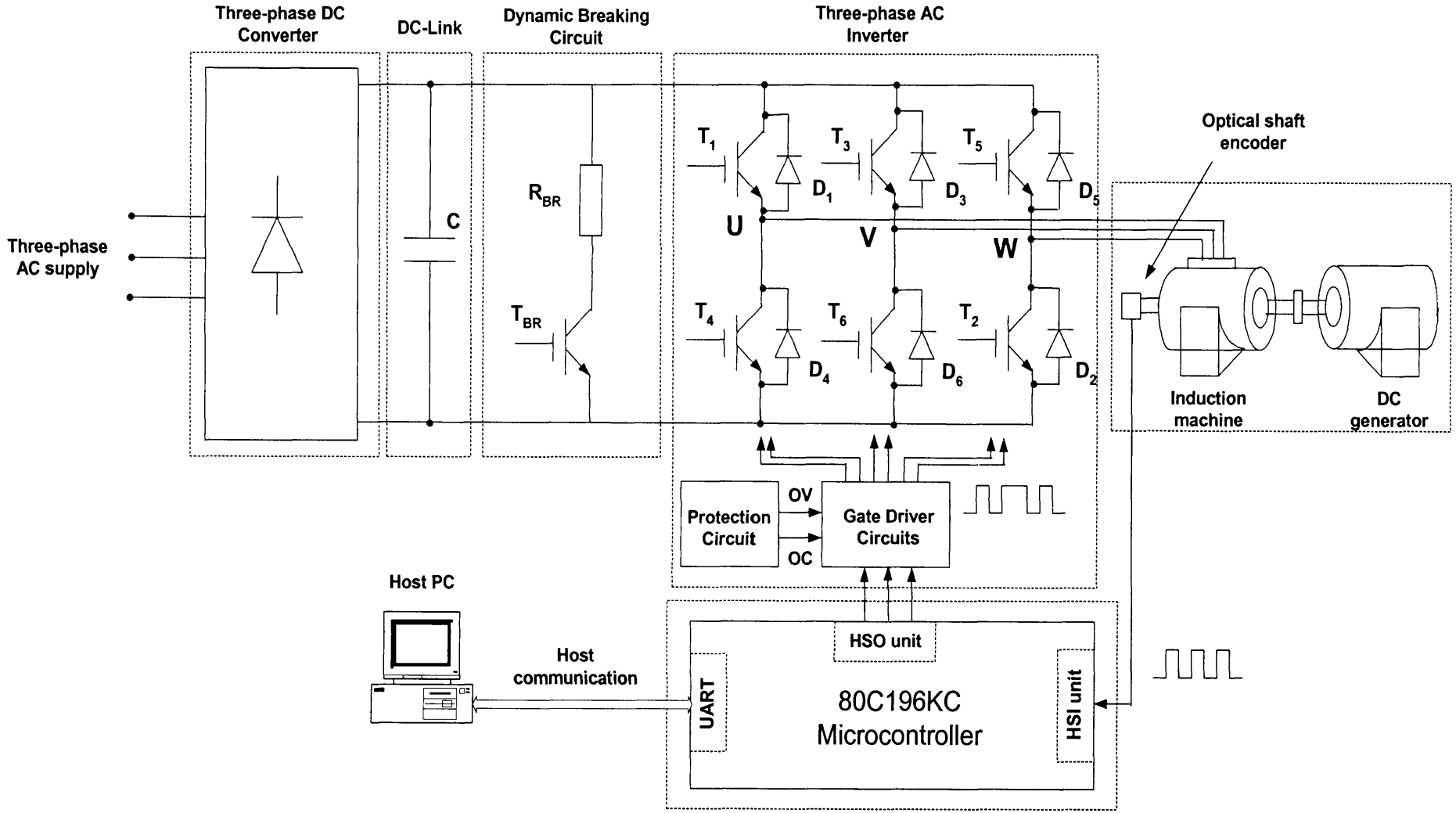


Figure 7-1: Simplified schematics of the hardware set-up for digital controlled IM drive system.

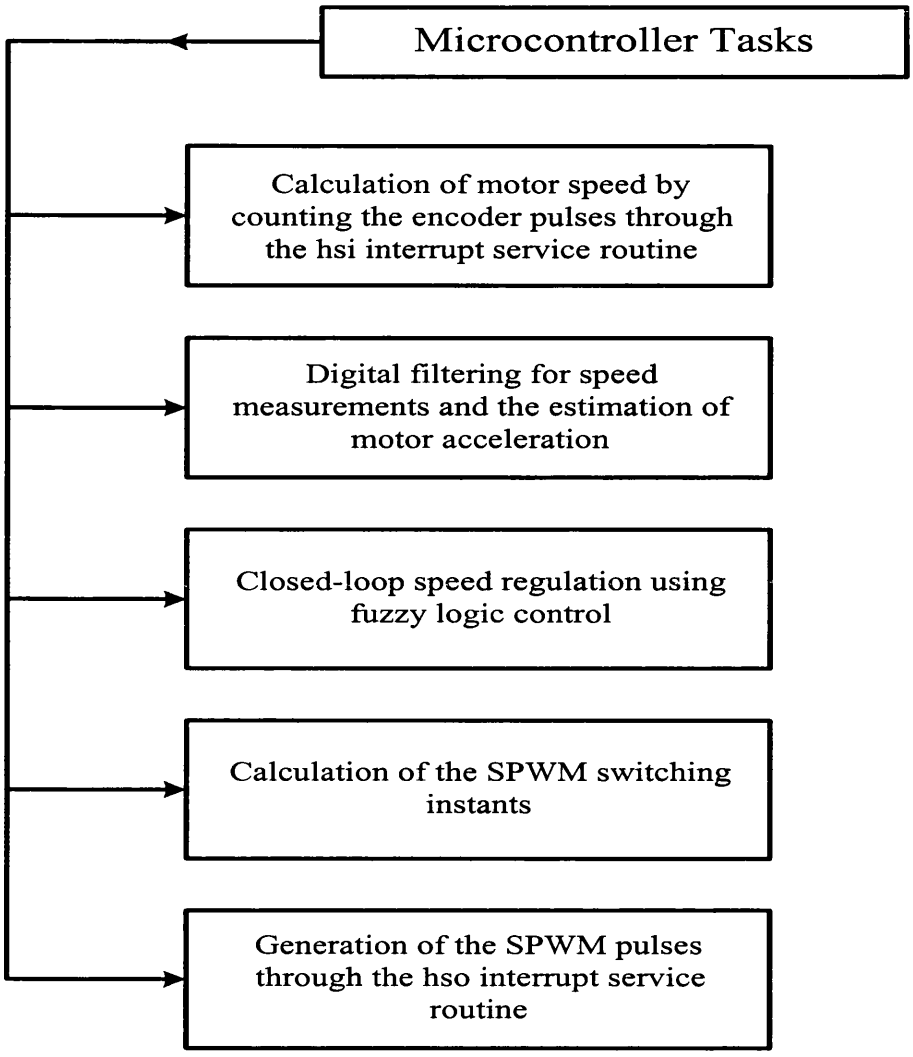


Figure 7-2: Main tasks of the 80C196KC microcontroller.

7.3 Digital speed measurement and digital filter design:

In speed drives application, it is greatly desirable to design a high accurate speed measurement method, in order to achieve an improved control performance of a speed drive system and make full use of the digital technology. The optical shaft encoder (also know as pulse generator) is often used for speed detection purposes in a microcontroller-based speed control environments. The superiority of the shaft encoder over the conventional speed detectors such as tacho-generators is realised in its ability to provide an effective solution for attaining a high performance and robust rotor speed measurements (Ohmae et al. 1982; Saito et al. 1984). As mentioned before, the encoder is

mechanically coupled to the motor shaft, which generates a train of pulses whose frequency is proportional to the motor speed. Since the generated pulses are digital in nature, the resulted output from the encoder can be fed directly to the microcontroller for speed computation. Various computation methods have been reported in literature for speed estimation (Ohmae et al. 1982; Saito et al. 1984; Saito et al. 1988; Guo 1994). The most commonly used are the T method, the M method and the M/T method. The latter is used in the present speed drive system, due to the inherent features of high resolution and high accuracy in a short detection time as reported by Ohmae et al. (1982). The basic principle of this method is presented in Appendix C, in which the computed variable of the rotor speed, ω_r, is expressed by the following formula,

$$\omega_r = \frac{60\, f_s\, M_1}{P_s\, M_2} \tag{7.1}$$

7.3.1 Digital implementation of the M/T method:

As mentioned before, the rotor speed is detected by a dual-phase pulse optical encoder with the two phases (A, B) apart by 90° and with a resolution of 1024 pulses per one mechanical rotation. In general, however, the accuracy of the speed measurement can be further improved with the augmentation of the encoder resolution. For this reason, the resolution of the used encoder can be doubled by feeding the two output signals from the encoder circuit to an Exclusive-OR (EX-OR) logic gate. Consequently, the resulted output would be a pulse train signal with twice the frequency of phase A or phase B. Therefore, the constant P_s in Equation (7.1) becomes 2048 instead of 1024. The basic operation of the EX-OR gate is given in Appendix C.

On the other hand, the frequency of the clock pulse, f_s, is defined by the clock frequency of *Timer_1*, which is clocked every 1 μs, and hence f_s equals to 1 MHz. Thus, the term $60\, f_s / P_s$ in Equation (7.1) is a constant. Moreover, the variable M_1 in the same equation is computed by using the external clocking feature of *Timer_2*. In other words, if the external clock is provided by the encoder pulses via HSI.1 pin on the microcontroller, then *Timer_2* counts on both positive and negative transitions of each pulse. It follows from this that the value recorded by *Timer_2* at the end of the speed detection period (T_d) is actually double the number of pulses that are generated from the EX-OR gate. Thus,

$$M_1 = \frac{Timer\_2}{2} \tag{7.2}$$

Furthermore, the value of M_2 for each consecutive detection period (T_d) is determined by subtracting the recorded timing value of *Timer_1* at the start of T_d from that recorded at the end. Referring to the timing diagram of Figure C.1, this can be expressed as,

$$M_2 = t\_end - t\_start \tag{7.3}$$

where *t_start* and *t_end* are the *Timer_1* clock timing values corresponding to times t_1 and t_3 (also shown in Figure C.1), respectively. Substituting Equations (7.2) and (7.3) into (7.1) yields the following formula,

$$\omega_r = \frac{60\ f_s\ Timer\_2}{2\ P_s(t\_end - t\_start)} \tag{7.4}$$

The digital implementation of Equation (7.4) can be utilised on the microcontroller by the succession use of the HSI.0 and SWT interrupt service routines. At the beginning of each detection period T_d, the HSI.0 interrupt occurs at first leading edge of the encoder pulse to record the initial timing value t_1 of *Timer_1*, which represents the value of *t_start*. This in turn provides synchronisation with the first generated encoder pulse. Once this interrupt occurred, *Timer_2* starts counting the leading and falling edges of the generated encoder pulses through HSI.1 pin on the microcontroller.

This is followed by disabling the HSI.0 interrupt and enabling the SWT interrupt which is utilised to set a constant sampling period T_s. In the present work, T_s is set to be equal to 5 msec. This interrupt is serviced once the prescribed sampling period T_s is expired (or reached) following the time value at t_1 which is captured by *Timer_1*. In this interrupt routine, the HSI.0 interrupt is enabled again to determine the timing value t_3 of *Timer_1*, which represents the end (*t_end*) of the detection period T_d. It should be noted that the value of *t_end* for this detection period will represent the *t_start* for the new detection period. The value of *Timer_2* is then recorded before it is reinitialised to be used again for a new detection period. Once the values of *t_start*, *t_end*, and *Timer_2* are defined, then Equations (7.4) is computed for the rotor speed of the induction motor. The operation of the HSI.0 and SWT interrupt routines are illustrated in the flowchart diagrams of Figure 7.3.

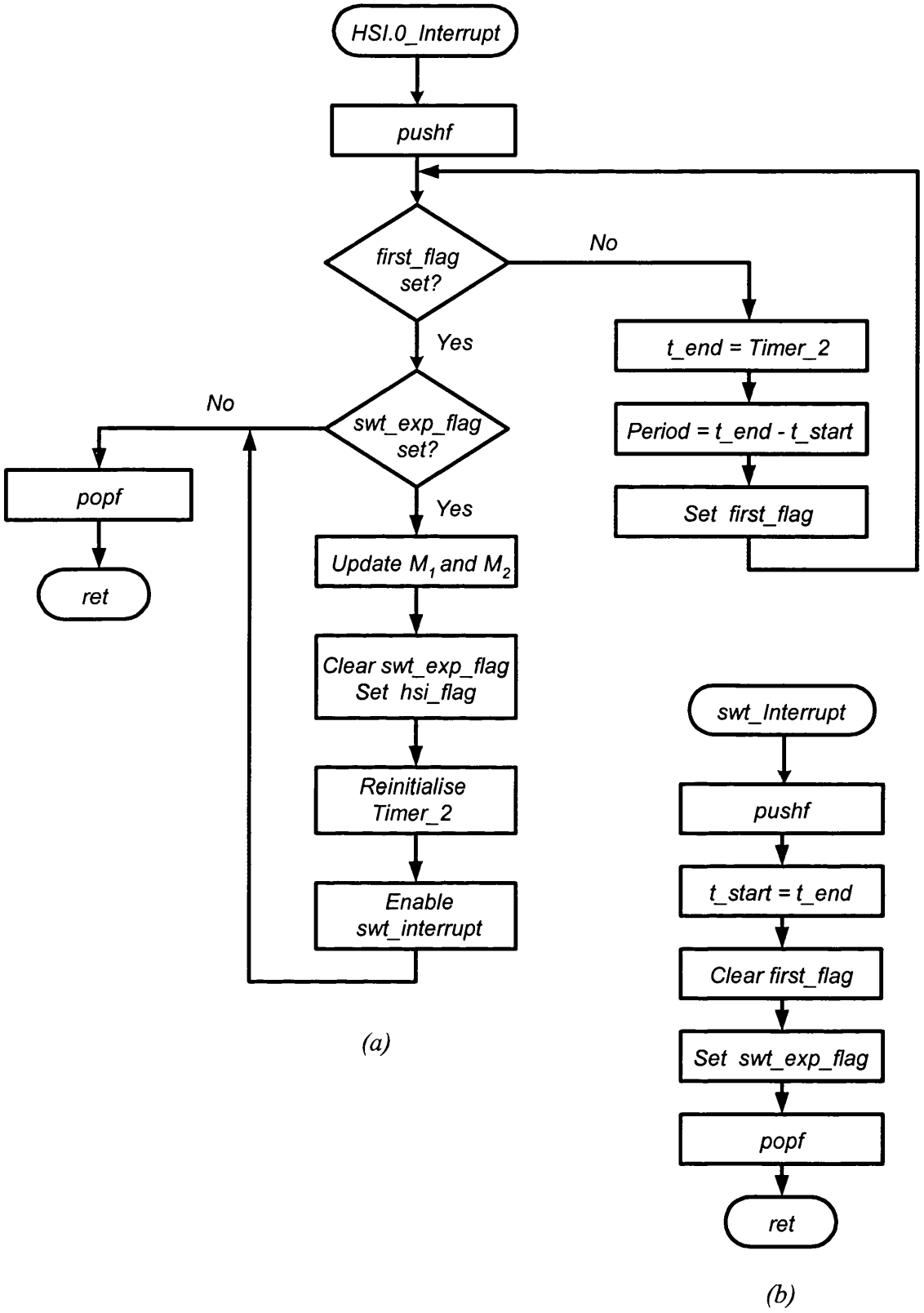


Figure 7-3: Flowchart diagram of the interrupt service routines used for digital speed measurement; (a) HSI.0 interrupt routine; (b) SWT interrupt routine.

7.3.2 *Smoothing techniques and digital filter design:*

A sufficient speed measurement (or estimation) technique is exceedingly needed in order to attain a high performance speed drives over a wide operating speed range especially at low speed. This is achieved by the digital implementation of the M/T method for a microcontroller-based system, as mentioned before in the previous section. Since the motor speed represents the significant element in the system's closed-loop operation, it is vital to obtain an instantaneous speed value that coincides with the true speed of the motor. Although speed estimation with high accuracy and resolution requirements can be obtained with the M/T method, its imperfection at lower speed range has been noted according to the investigation done by Guo (1994).

Furthermore, the motor speed is commonly detected by counting the generated pulses from the shaft encoder within a prescribed time interval (detection period T_d). Thus, the possibility of having an incorrect number of pulses within the time T_d is high. This is due to the time needed for multi-tasking computation on the microcontroller-based system, which affects the response time of the HSI.0 interrupt routine. Moreover, the existence of the electromagnetic interference (EMI) noise can influence adversely the performance of both the microcontroller and the shaft encoder systems. This noise is created due to the rapid speed changes of the induction motor which might causes sudden changes of currents and induced voltages in the stator and rotor windings (Keiser 1981). Also, variations in rotor speed inevitably occur due to the use of static power devices (IGBTs), which draw current in switching pulses during the modulation cycle.

Several methods have been proposed by Saito et al. (1988) to tackle the aforementioned problems with speed measurements. However, these methods require multiplication and division operations, which virtually increase the overheads of the microcontroller processing time. This led to the use of a simple estimation method for the purpose of digital filtering known as the Moving Average Filter (MAF) (Brown 1963; Cunningham 1992). This filter is basically a running estimate of the mean value, which simply involves averaging a sequence of data of length N in successive sub-sequences of length $m < N$, thus earning the name moving average (also known as rolling average). This can be defined by the following expression,

$$y_k = \frac{1}{m} \sum_{i=0}^{m-1} x_{k-i}, \qquad m-1 \le k \le N \tag{7.5}$$

where m defines the number of samples (or observations) which represents the number of the computed speed values to be averaged at each successive detection time T_d. In effect, the performance of this filter depends significantly on the value of m. In other words, the value of m is usually selected high for a stable and accurate estimate of the averaged speed value. But, however, this would have an ill effect on the drive system performance in terms of control response. In the present work, therefore, a value of 4 is chosen for m, in order to attain a near accurate estimate of the averaged speed value with rapid system response. It follows from this that the computing of Equation (7.5) can be accomplished by only using add and shift operations which requires a shorter processing time with no compulsions to multiplication and division operations.

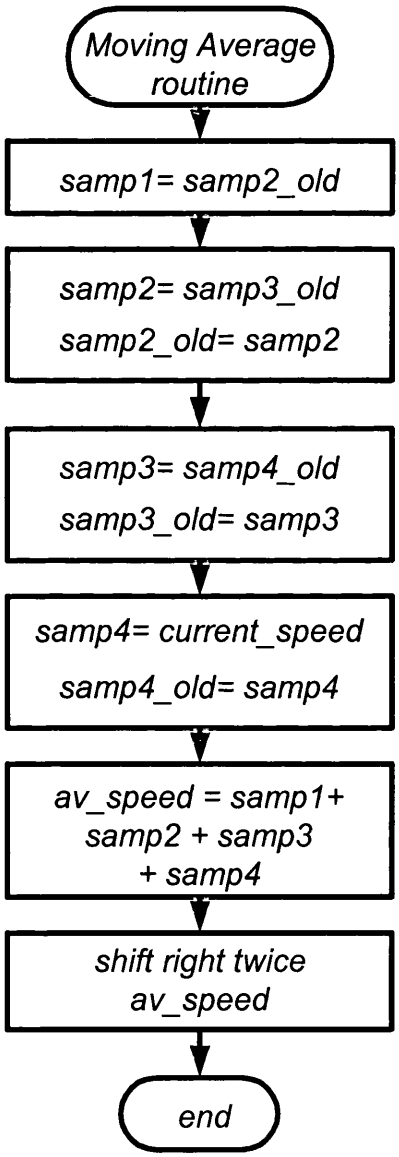


Figure 7-4: Flowchart diagram of the real time moving average routine.

In the light of the above considerations, the moving average method can be regarded as a simple low-pass filter that passes a single frequency component at the mean value with a cut-off frequency given by,

$$\omega_c = \frac{\omega_{samp}}{2\,m} \tag{7.6}$$

where ω_{samp} denotes the sampling frequency, which is equal to the reciprocal of the sampling time T_s. Therefore, ω_{samp} is equal to 200 Hz and if m is set to 4, then the cut-off frequency ω_c becomes 25 Hz. The principle of operation of the moving average method along with the procedures of the filter design are presented in Appendix D. Figure 7-4 shows a simple flowchart diagram which summarises the basic operation of the real time moving average routine.

7.4 Calculation of switching instants for variable frequency, variable voltage SPWM waveforms:

In the present work, the two level regular asymmetric SPWM technique, given in Chapter 2, is implemented on the microcontroller in order to generate near-sinusoidal current waveforms at the output of the VSI bridge inverter. In this technique, the modulating signal is sampled twice at equal time intervals every carrier period T_c, as shown in Figure 2-9. The switching instants of the pulses are defined by solving the following equations for the trailing and leading edges of the carrier wave,

$$t_{on}(k) = T_c\,k + \frac{T_c}{4}\left[1 - M_r \sin(k\,T_c)\right] \tag{7.7}$$

$$t_{off}(k) = T_c\,k + \frac{T_c}{4}\left[3 + M_r \sin\left(k\,T_c + \frac{T_c}{2}\right)\right] \tag{7.8}$$

where the variable k represents the number of pulses in the modulating cycle, which is equal to the value of frequency ratio f_r. Using the above equations, the time intervals of the on and off sequences for each pulse are determined. Nevertheless, the computation time of these intervals can be reduced significantly using the intrinsic symmetry features of the PWM waveforms. In other words, if odd

values of f_r are used, then the PWM signal becomes a half wave symmetrical signal, that solely contains odd harmonics. Thus, it is only essential to compute the switching instants of the pulses for the first half of the modulating cycle and use the symmetry of the PWM waveforms to define the switching instants of the remaining pulses in the second half of the cycle. It should be noted that, the computations of t_{on} and t_{off} in Equations (7.7) and (7.8) strictly require the knowledge of both the modulation ratio M_r and the carrier period T_c. The latter is the reciprocal of the carrier frequency f_c, which is defined by computing the arithmetic product between the modulating frequency f_m and the frequency ratio f_r. The value of the modulating frequency f_m corresponds to the value of the stator frequency command $\omega_s^*(t)$, which is generated from the speed controller as illustrated in the preceding chapter. Similarly, the value of the modulation ratio M_r coincides with value of the desired inverter voltage V_s^*, which is defined from $\omega_s^*(t)$ by the use of simple function generator. Therefore, for a constant *V/f* operation, the modulating frequency f_m is bound to be linearly related to the modulation ratio M_r in agreement with the following equation,

$$M_r = d f_m \tag{7.9}$$

On the other hand, the sin terms in Equations (7.7) and (7.8) characterise the sinusoidal modulating signal of the PWM waveforms at certain time instant defined by the value of kT_c. Since the on line computation of these terms on the microcontroller requires a long processing time which might adversely effects the response of the control system, an alternative approach is highly required. If the frequency ratio f_r is kept constant throughout the signal modulation, then the value of the carrier period T_c is directly related to the modulating frequency f_m.

Thus, only the variation of the carrier period T_c (in units of microseconds) is needed to perform the *V/f* control for a fixed value of frequency ratio f_r. Consequently, a fixed value of T_c in degrees can be attained for the sin terms by dividing the value of full modulating cycle (360°) over the frequency ratio. As a result, it is required to compute the sin values for all the switching instants only once, over the whole modulating cycle. These computations, therefore, can be conveniently accomplished off-line and the results stored as a look-up table prior the commencement of the switching instants calculations. This table is labelled as "sin_table" in the system program, and stored in the external RAM of the microcontroller at the starting address of 5500h.

Having acquired the values of M_r and T_c and using the look-up sin table, the switching instants t_{on} and t_{off} are then calculated using Equations (7.7) and (7.8), respectively, for the first half cycle. These calculated instants are then stored in a dynamical table, which is updated at every sampling time, T_s. Once this table is completed, the data stored are then swapped to another table in order to be used in the HSO interrupt service routine. Thus, two dynamical tables stored in the external RAM are consecutively used for the purpose of generating the PWM waveforms. The first "inoperative- table" is used to store a new set of computed switching instants corresponding to any changes in the inverter frequency and voltage commands, while the second "operative-table" includes the previously calculated switching instants which are currently used in the interrupt routine for PWM generation. The swapping process of the two tables is mainly performed at the end of the currently generated cycle in the interrupt routine depending on the state of a "swap_flag".

7.5 HSO interrupt routine for generating three phase SPWM waveforms:

The HSO unit that is embedded on the microcontroller is programmed to facilitate the generation of the three phase PWM waveforms through a complete interrupt control. For this purpose, three HSO channels (HSO.0 through HSO.2 pins) are being used to output three sinusoidal PWM signals with a phase shift of 120°, which each corresponds to one stator phase of the induction motor, respectively. The time values of the switching instants for these pulses are given in the operative-table which contains the time values for a complete half cycle. These signals are then passed through a gate driver circuit to produce the necessary six firing PWM signals for the three phase bridge inverter, as shown in Figure 7-1.

To ensure a phase shift of 120° between the generated PWM signals, three pointers (*Pointer_1*, *Pointer_2*, and *Pointer_3*) are being used, each of which points at a certain time value in the operative-table with a phase shift of one-sixth of a cycle between them. It means that *Pointer_1* points at time values between 0° to 60°, *Pointer_2* points at time values between 60° to 120°, while *Pointer_3* points at time values between 120° to 180°. The three complete PWM signals are then produced using the symmetry features of the sinusoidal PWM signals alongside the designed pointers. Thus, the generation of the three phase PWM waveforms is accomplished using the same time values stored in the operative-table but their loading in the HSO unit is carried out in either forward or reverse switching order.

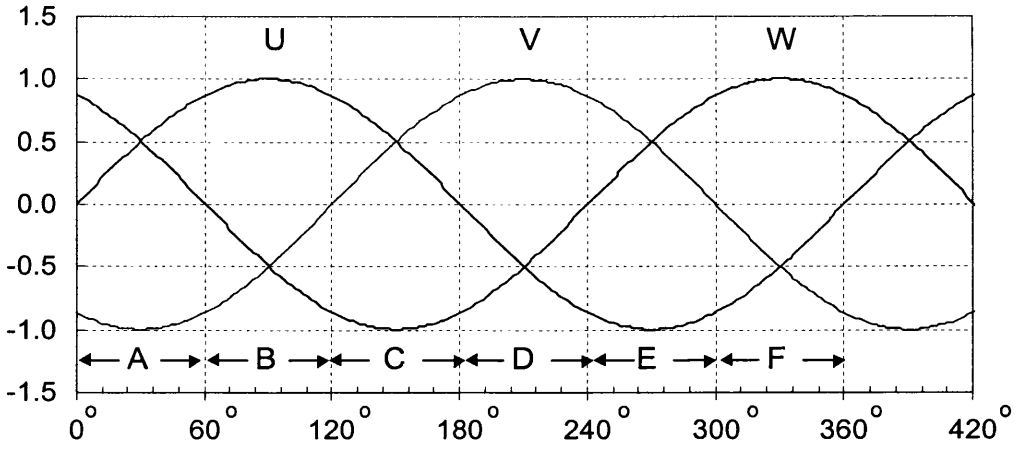


Figure 7-5: The symmetry of the three phase sinusoidal waveforms.

In other words, the difference between generating the positive and negative parts of a cycle is only based on the type of switching event. It means that, if the HSO channel is turned on at certain time instant brought from the operative table in the positive part of the cycle, then the same time instant with a phase shift of 180° represents a turning off for the same channel in the negative part of the cycle, and vice versa. Thus, the forward and reverse switching orders of the three implied HSO channels occur in the positive and negative parts of the cycle, respectively. This can be clearly clarified by referring to Figure 7-5. As shown in this figure, the complete 360° cycles for the three phases (*U*, *V*, *W*) are divided into six segments labelled as "*A*, *B*, *C*, *D*, *E*, *F*", each of which spans for one-sixth of a cycle.

Based on this approach, only one look-up table is needed to produce the full three phase PWM waveforms. In segment *A*, for example, *Pointer_1* and *Pointer_3* are used to load the time values for phases *U* and *W* in forward switching order, respectively, while *Pointer_2* is used to load the time values for phase *V*, but in the reverse switching order. Considering a 180° phase shift (segment *D*), *Pointer_1* and *Pointer_3* are still used for phases *U* and *W*, but they in the reverse switching order. Similarly, *Pointer_2* is still used for phase *V*, but in the forward switching order. The same switching arrangements are then sequentially applied for the rest of the segments. Table 7.1 shows the switching sequences used to generate the three phase PWM pulses in each segment, in which phase *U* is considered as a reference waveform. The interchange between these switching segments in the interrupt routine is performed at the beginning of every one-sixth of a cycle using a pre-defined flag called "*per_flag*". Thus, based on the state of this flag, the appropriate switching segment is selected.

Phases / Segments	*U*	*V*	*W*
A (0 – 60°)	*Pointer_1* (Forward)	*Pointer_2* (Reverse)	*Pointer_3* (Forward)
B (60° - 120°)	*Pointer_2* (Forward)	*Pointer_3* (Reverse)	*Pointer_1* (Reverse)
C (120° - 180°)	*Pointer_3* (Forward)	*Pointer_1* (Forward)	*Pointer_2* (Reverse)
D (180° - 240°)	*Pointer_1* (Reverse)	*Pointer_2* (Forward)	*Pointer_3* (Reverse)
E (240° - 300°)	*Pointer_2* (Reverse)	*Pointer_3* (Forward)	*Pointer_1* (Forward)
F (300° - 360°)	*Pointer_3* (Reverse)	*Pointer_1* (Reverse)	*Pointer_2* (Forward)

Table 7.1: Switching sequences to generate three phase PWM pulses.

The HSO interrupt is enabled by setting the third bit of the interrupt mask (INT_MASK) byte register and then forced to occur by setting the third bit of the interrupt pending (INT_PEND) byte register. After executing these commands, the program enters a wait loop, awaiting the HSO interrupt to occur. Once there is no higher priority interrupt pending, the microcontroller acknowledges this interrupt and then services the HSO interrupt routine. The interrupt routine starts by pushing the program counter into the stack using the PUSHF command, and execution continues with the first instruction in the interrupt service routine.

Every time the microcontroller enters the HSO interrupt routine, two time values brought from the operative table for each phase are loaded into the CAM register. These time values are then compared with *Timer_1*, and when a match occurs an event of turning on or off is produced on the corresponding HSO channel. The three pointers (*Pointer_1*, *Pointer_2*, and *Pointer_3*) are then incremented to start pointing at the next time values to be loaded into the CAM. In addition, a programmed counter called "*counter1*" is incremented by one, each time the microcontroller does service the HSO interrupt routine.

Before leaving the interrupt routine, the value stored in this counter is then compared with "*P_third*" register, which contains the number of pulses in one-sixth of a cycle. If it is less than *P_third*, the program exits the interrupt routine, so that the interrupted program resumes its operation. This process will carry on until the value of *counter1* is equal to the pre-defined value of

P_third. Once this happens, *counter1* is reinitialised with the value of zero, the three pointers are readjusted to their starting addresses on the operative table, and *per_flag* is set so that to start generating PWM pulses in the following switching segment the next time the HSO interrupt routine is called. This sequence of operation is repeated for all the remaining switching segments until the three phase PWM waveforms are generated. All these operations are summarised in the flow chart of the HSO interrupt service routine, shown in Figure 7-6. It should be noted that, the set of operations encompassed by the dashed green lines in the flow chart is repeated for the remaining switching segments in succession. However, after generating the last PWM pulse in the final switching segment "*F*", table swapping occurs and "swap_flag" is set in order to inform the main program that table swapping has occurred.

7.6 Real time implementation of closed loop fuzzy logic speed control system:

The developed fuzzy logic control system for *V/f* speed regulators as described in the foregoing chapter is implemented on the microcontroller. Some modifications have been made to simplify the fuzzy logic algorithm in order to ease its implementation, as later presented in this section. Before reaching that point, certain tasks have to be considered regarding the experimented speed control system. One of these is the deployment of a soft-starting operation to overcome the drawbacks of direct closed-loop starting of the induction motor from the stall position, such as high starting currents and stator voltage peaks. To achieve this, the stator frequency and voltage are both initialised with reduced values at the starting mode in such away to produce sufficient current and torque to meet the requirements of the motor driven load.

From this starting value, the stator frequency (considering a *V/f* control) is smoothly increased to a pre-set value before commencing the closed loop operation. Thus, the modulating frequency f_s is set initially to be equal to 62 rad/sec, which is equivalent to 10 Hz. This frequency is then augmented by a value of 6 rad/sec at the end of each computed modulating cycle, until the value of 90 rad/sec is reached. This is in turn will bring gently the induction motor to a desired speed value before actually applying the closed loop operation and hence simplifying the step response tests of the fuzzy logic controller. In addition, the switching between soft-starting and closed loop modes is made simple by connecting a preset to Channel 4 of the on chip A/D converter. Hence, mode selection is accomplished according to the value given by the preset.

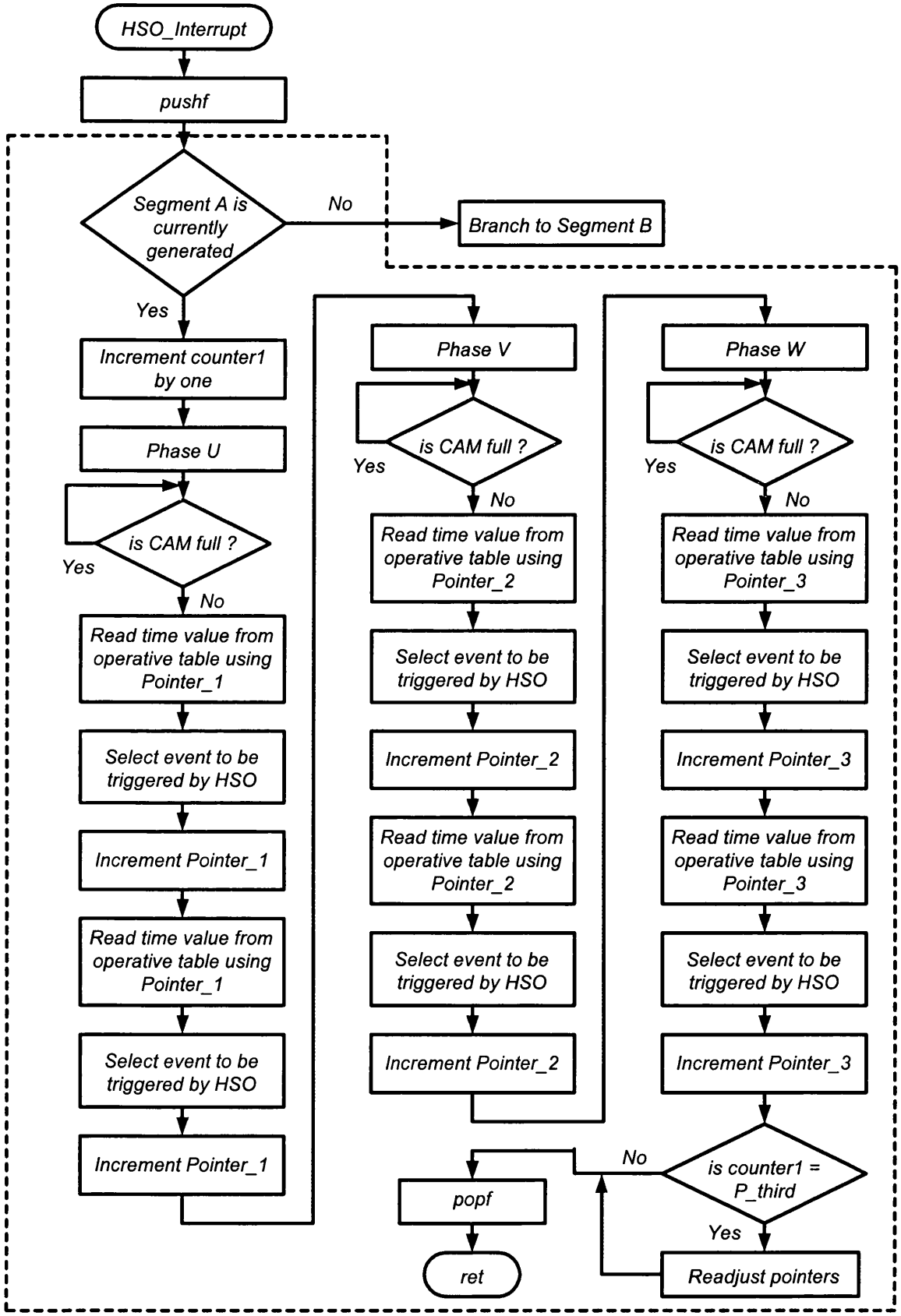


Figure 7-6: Flowchart diagram of the HSO interrupt service routine.

				CSEG at 5000h		*Fuzzy_table*:							
dcw	0	19	38	38	38	47	59	59	59	63	77	78	78
dcw	-19	0	19	19	38	47	47	48	59	63	63	63	78
dcw	-38	-19	0	19	38	38	38	48	59	59	59	63	78
dcw	-38	-19	-19	0	19	19	38	47	47	47	59	63	63
dcw	-38	-38	-38	-19	0	19	38	38	38	47	59	59	59
dcw	-48	-48	-37	-19	-19	0	19	19	38	47	48	48	59
dcw	-59	-48	-37	-37	-37	-19	0	19	38	38	38	47	59
dcw	-59	-48	-48	-48	-37	-19	-19	0	19	19	38	47	47
dcw	-59	-59	-59	-48	-37	-37	-37	-19	0	19	38	38	38
dcw	-63	-63	-59	-48	-48	-48	-37	-19	-19	0	19	19	38
dcw	-78	-63	-59	-59	-59	-48	-37	-37	-37	-19	0	19	38
dcw	-78	-63	-63	-63	-59	-48	-46	-46	-37	-19	-19	0	19
dcw	-78	-78	-78	-63	-59	-59	-59	-46	-37	-37	-37	-19	0

Table 7.2: Fuzzy look-up decision table as stored on the microcontroller.

In the proposed closed-loop scheme, the inputs to the fuzzy logic control system are the speed error e_o and its rate of change Δe_o, while the output is the slip frequency change $\Delta\omega_{sl}^*$. As presented in Chapter 4, the controller output is generated according to some linguistic rules, which are designed based on control engineering knowledge of the drive system. Thus, for a given speed error and its rate of change, the control rules have been developed and listed in Table 4.1 in Chapter 4. In addition, the input and output variables of the FLC system have been discretised into thirteen quantised levels for the purpose of microcontroller implementation.

Consequently, the look-up decision table, shown in Table 6.2 (Chapter 6), was constructed from Table 4.1 and Table 6.1 using the simulated fuzzy system model as developed in Chapter 4. This table is labelled as "*fuzzy_table*" in the system program, and stored in the external RAM of the microcontroller, as shown above in Table 7.2. Here, "*dcw*" specifies word variables and CSEG means that they are stored starting at the given address. On the other hand, the following tasks are carried out in order to implement the fuzzy logic control system on the microcontroller.

1) Computing the inputs of the fuzzy logic controller (FLC), which are the speed error e_o and its derivative Δe_o. The speed error is defined by subtracting the instantaneous measured speed of the induction motor from the desired reference speed, while its derivative is defined by subtracting the previously computed speed error from the currently defined error. In addition, they can have positive and negative physical values due to acceleration and deceleration of the induction motor. The handling of such values can be easily performed on the microcontroller using signed or unsigned integer arithmetic (16-bit embedded microcontroller 1990).

2) Based on the defined values of speed error and change of error, the corresponding incremental slip frequency command $\Delta\omega_{sl}^{*}$ is selected from Table 7.2. The selection is made using a set of comparison operations between the actual values of speed error and change of error and the corresponding reference values. The partitions of the positive and negative reference values for both the speed error and the change of error is shown in Figure 7-7, with which the appropriate address for the corresponding slip frequency in Table 7.2 is determined. The flow chart diagram of the comparison operations is shown in Figure 7-8.

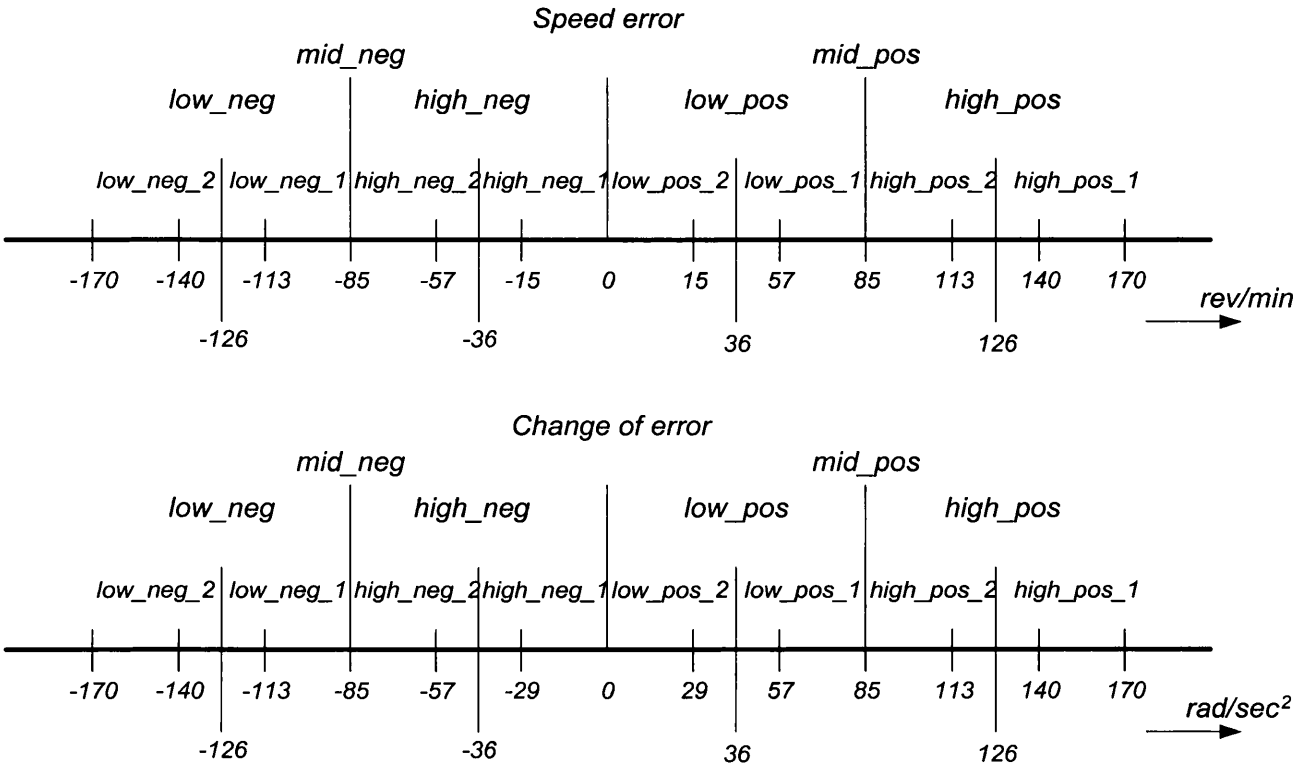


Figure 7-7: Partitions of reference values; (a) Speed error, (b) Change of error.

3) Computing the integration of the output from the FLC system in digital form using the following integration formula,

$$\omega_{sl}^{*}(k) = \Delta\, \omega_{sl}^{*}\,(k)\ *\ T_s + \ \omega_{sl}\,(k-1) \tag{7.10}$$

where T_s is the sampling time, and $\omega_{sl}\,(k-1)$ is the computed slip frequency at the preceding sample.

4) The new computed slip frequency is then aggregated to the rotational frequency of the motor shaft to attain a new stator frequency and its corresponding stator voltage, which are then passed to a subroutine in the main program to calculate the new switching instants of the PWM waveforms.

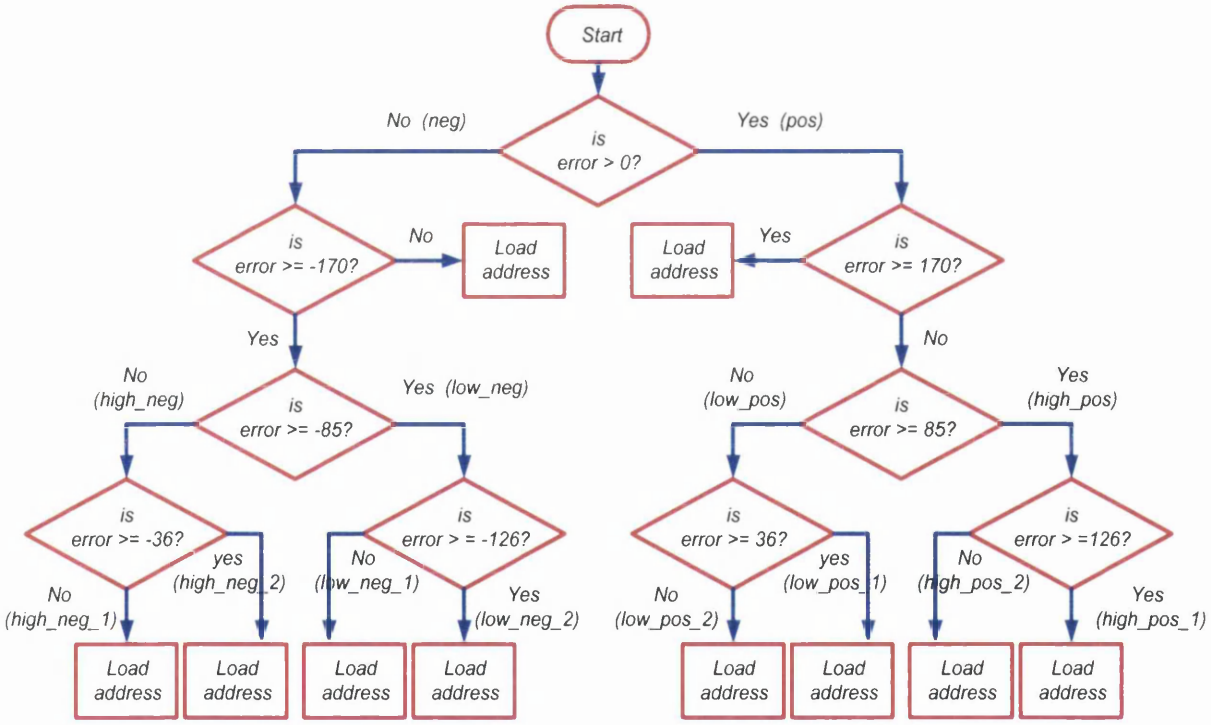


Figure 7-8: Flowchart diagram of the comparison operations for the FLC system.

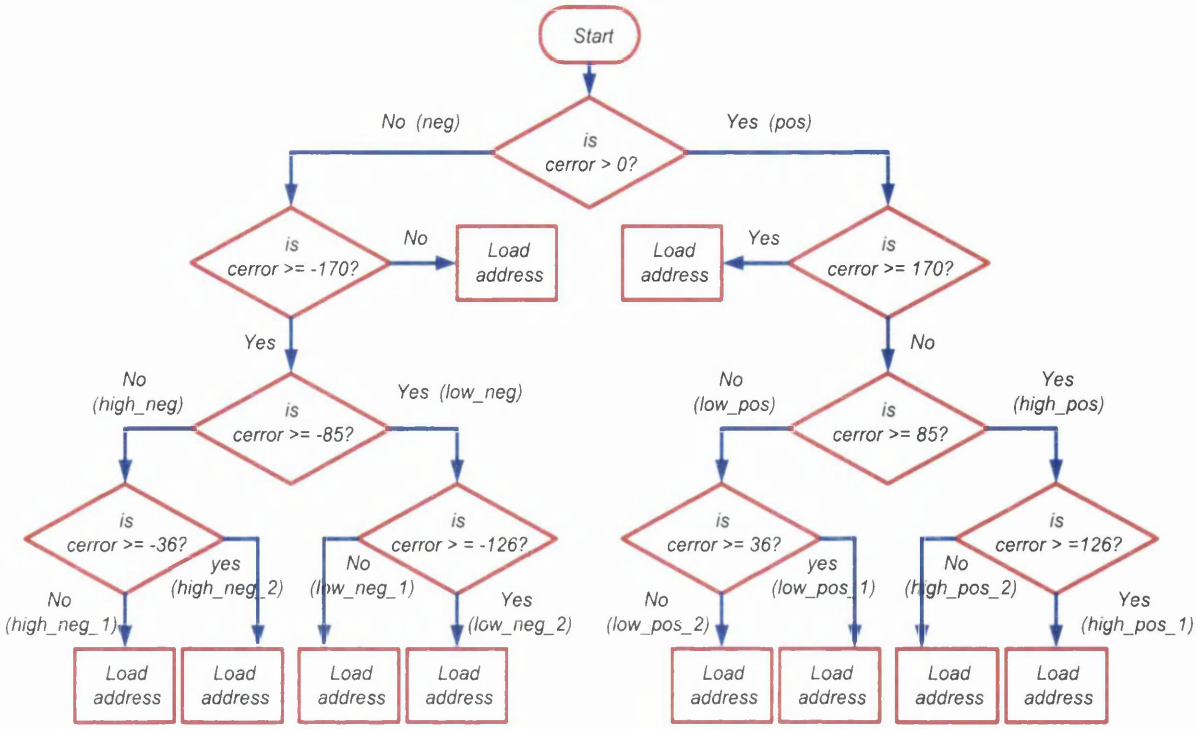


Figure 7-8: Continued

7.7 Summary:

In this chapter, general description of the experimental set-up for the FLC speed drive system was exhibited. It was followed by a detailed presentation of the software development for the closed loop speed control operation. It included the generation of the three-phase sinusoidal PWM waveforms and the techniques of digital speed measurements. The design methods for a digital filter based on the rolling average concept to improve the speed measurement were presented. In addition, flowcharts were given to demonstrate the operation of the software developed.

Chapter 8, Practical results and discussion:

8.1 Introduction:

In this chapter, practical results of the proposed variable speed drive system are presented and analysed to verify the validity of the simulation results obtained in the anterior chapters (Chapters 3 and 6). The rated parameters of the tested induction motor were determined experimentally and are given in Table 3.1. Comprehensive description of the experimental set-up for the implied speed drive system and software development for the closed-loop speed control operation are exhibited earlier in Chapter 7. The developed software is implemented on the microcontroller system portion, in which sets of control tasks (given in Figure 7-2) are performed. Some of these tasks such as the generation of PWM signals and digital speed measurement approach are individually developed and tested as separate code blocks before they are actually integrated together to collectively make up a complete source code.

The chapter starts with a brief investigation of the characteristics of some of the motor variables when the induction motor is subjected to different types of sinusoidal PWM signals for a modified switching frequency, f_s. In addition, the performance of the induction motor under open-loop control test is also presented. This is followed by illustrating the dynamic performance of the proposed speed control system, which demonstrates the validity of the designed FLC speed controller. Comparison studies between the acquired experimental results with that obtained via simulation, at the same operating conditions, are also considered.

8.2 Experimental PWM waveforms:

In this section the steady state performance of the induction motor are investigated when it is supplied by three-phase sinusoidal PWM (SPWM) voltage waveforms. For this purpose, the line current and its frequency spectrum, the line-to-line voltage and the phase voltage waveforms of the induction motor are shown. The conventional *Regular Asymmetric (RA)* SPWM technique, the *Third-Harmonic Injection (THI)* SPWM technique and the *Modified* SPWM are all employed in the present experimental tests. Detailed illustration of these types of switching techniques has been discussed in Chapter 2. For the purpose of testing these techniques, the DC link voltage of the

inverter is set to be equal to 80V DC, while the modulating frequency f_m of the PWM signal is chosen to be 50Hz. The frequency ratio f_r is selected to be 18 and 60 which correspond to 900Hz and 3kHz of switching frequency f_s, respectively. Figures 8-1, 8-2, and 8-3 show the intrinsic features of the *RA-SPWM*, the *THI-SPWM*, and the *Modified SPWM*, respectively, at a frequency ratio equals to 18. Table 8.1, shows the measured values of the phase and line voltages of the induction motor for the three implemented PWM switching types.

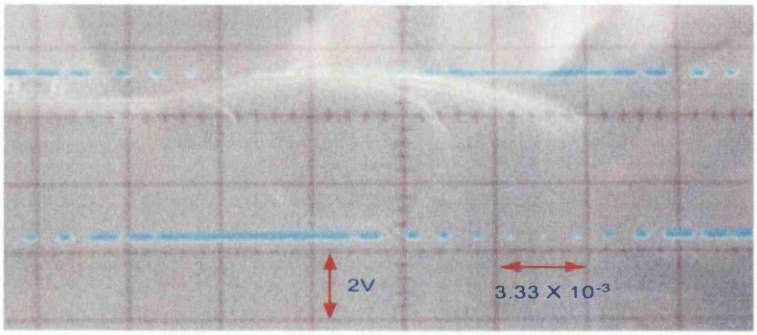
	Phase Voltage (V_{RMS})	Line-to-line Voltage (V_{RMS})
RA-SPWM	33 V	57.3V
THI-SPWM	35V	62V
Modified-SPWM	39V	67.5V

Table 8.1: Measured motor variables for the three PWM switching types at f_r =18.

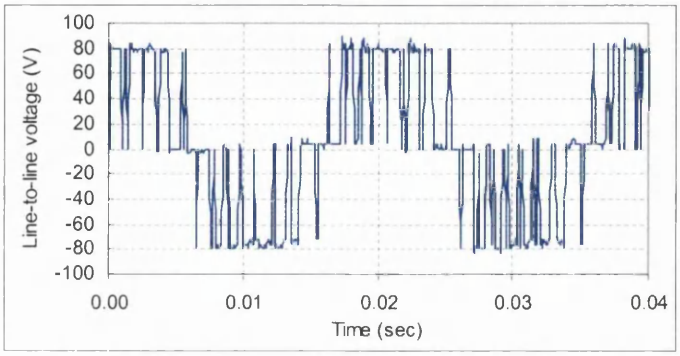
Inspection of these results shows that improved features of the conventional *RA-SPWM* waveforms can be attained with the employment of the *THI-SPWM* technique in terms of the resulted output voltage, while maintaining low distortion in the stator current. In addition, the *Modified-SPWM* technique can also provide further improvement to the output results, in which the generated line-to-line voltage is higher compared to that obtained with *THI-SPWM*. Furthermore, the *Modified-SPWM* technique produces less switching pulses in a cycle, which provides less switching stresses on the power devices, and thus reducing the switching losses. However, inspection of the current waveform in Figure 8-3 shows an increase of current ripple due to the unmodulated interval between 60° and 120° in the positive half of the cycle (also between 240° and 300° in the negative half-period).

On the other hand, the ripples in the line current waveforms are fairly reduced by increasing the value of the frequency ratio. This causes the predominant harmonics to be pushed into a higher frequency range, and hence permitting smoother current signal. Figures 8-4, 8-5, and 8-6 show the intrinsic features of the *RA-SPWM*, the *THI-SPWM*, and the *Modified SPWM*, respectively, at a frequency ratio equal to 60. Inspection of these figures shows that the generated ripples due to the harmonic contents in the waveforms are reduced with the increase in the value of the frequency ratio.

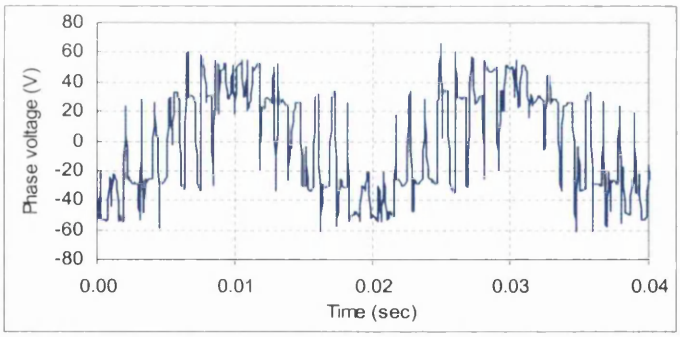
(a)



(b)



(c)



(d)

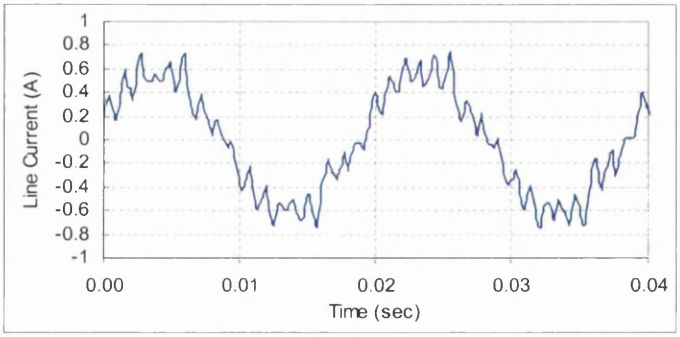
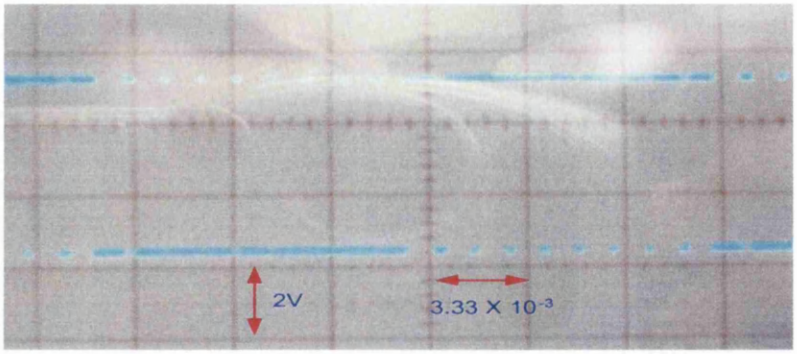
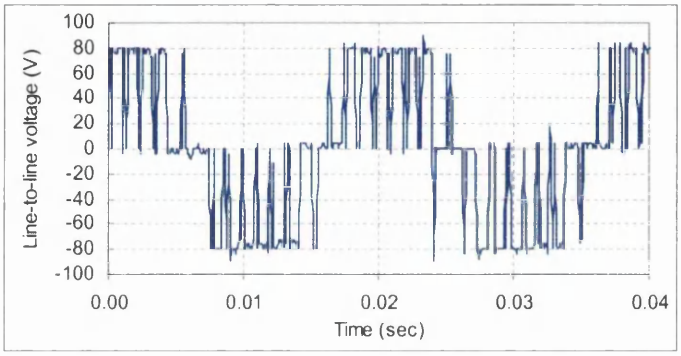


Figure 8-1: Experimental waveforms for RA-SPWM at f_r =18; (a) PWM gating signal; (b) line-to-line voltage;(c) phase voltage; (d) line current

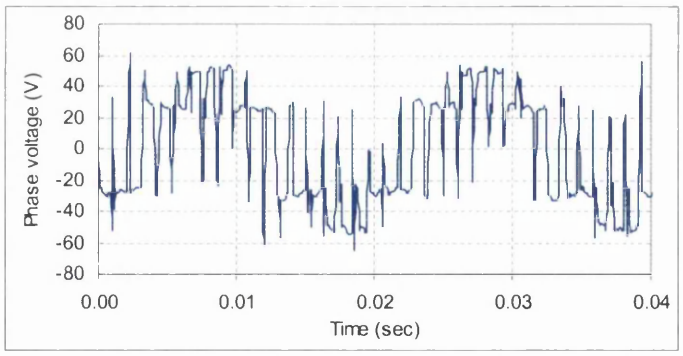
(a)



(b)



(c)



(d)

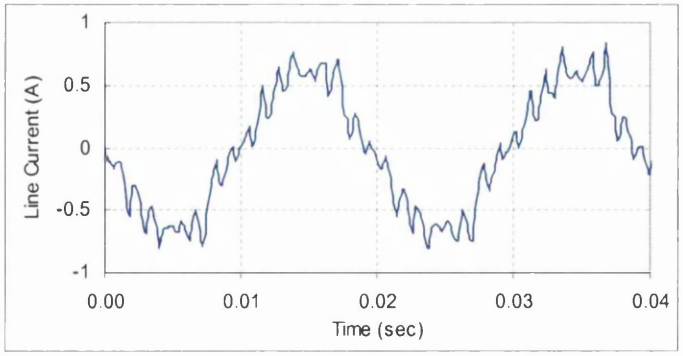
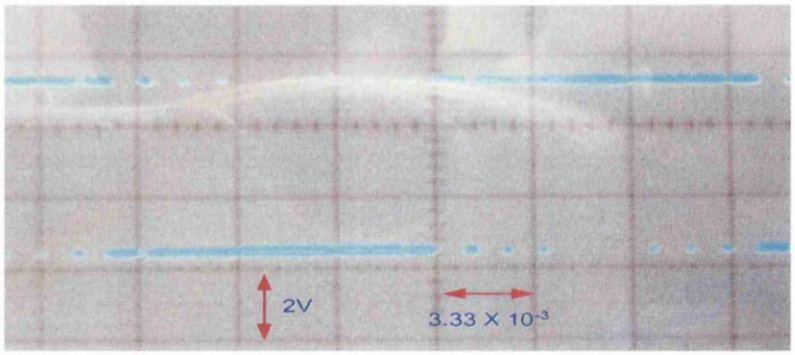
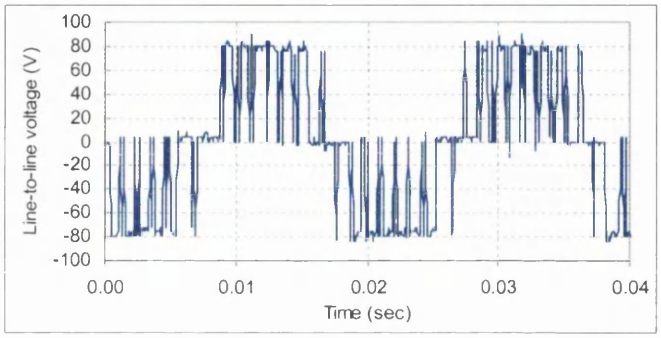


Figure 8-2: Experimental waveforms for THI-SPWM at f_r =18; (a) PWM gating signal; (b) line-to-line voltage;(c) phase voltage; (d) line current.

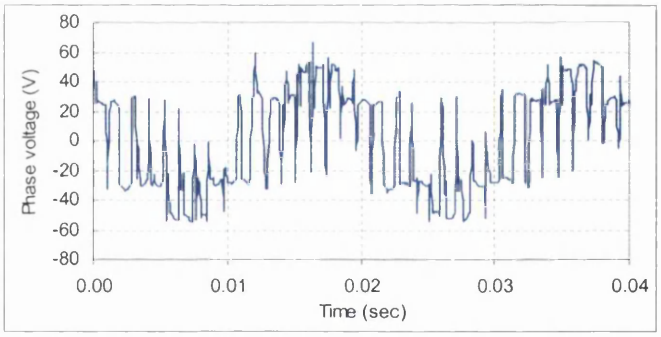
(a)



(b)



(c)



(d)

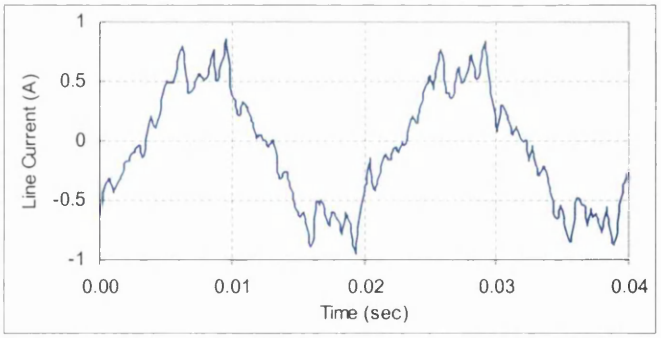
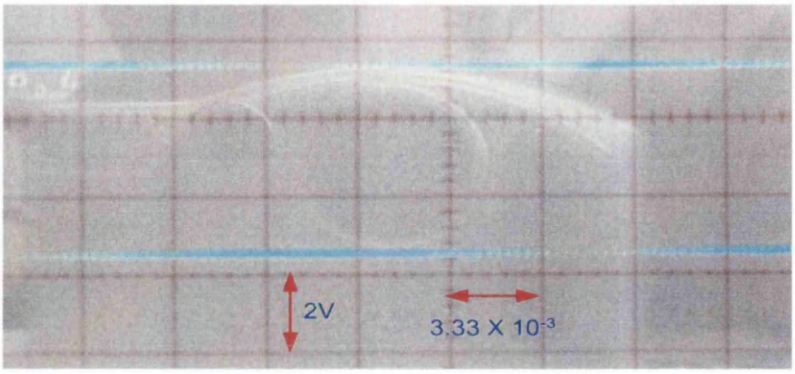
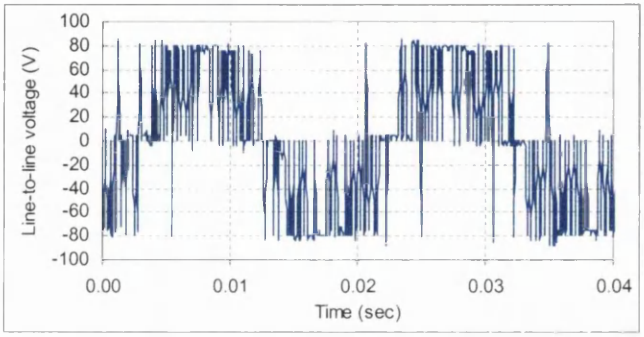


Figure 8-3: Experimental waveforms for the Modified-SPWM at f_r =18; (a) PWM gating signal; (b) line-to-line voltage;(c) phase voltage; (d) line current.

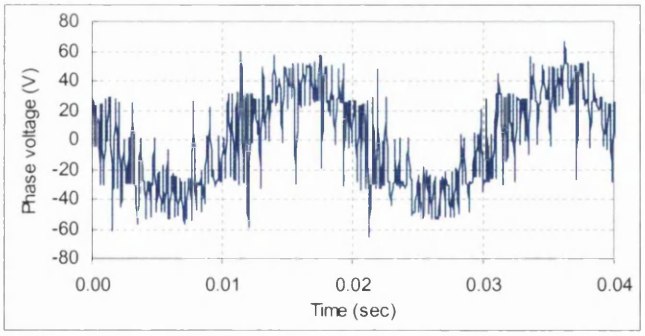
(a)



(b)



(c)



(d)

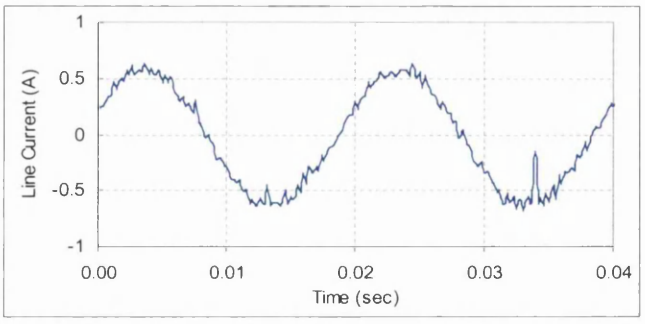
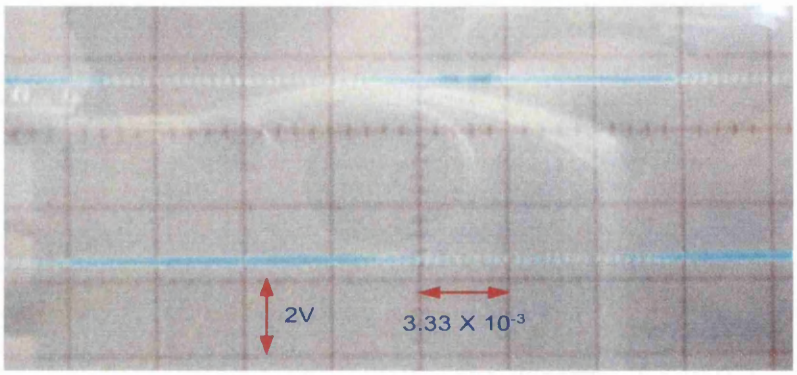
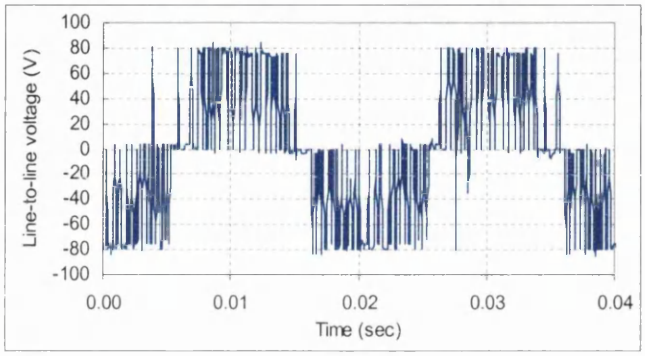


Figure 8-4: Experimental waveforms for the ARS-SPWM at f_r =60; (a) PWM gating signal; (b) line-to-line voltage;(c) phase voltage; (d) line current.

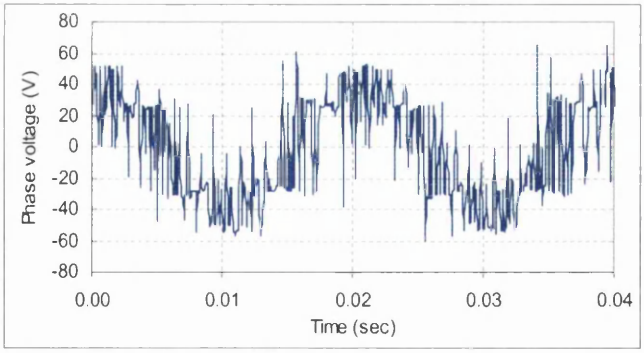
(a)



(b)



(c)



(d)

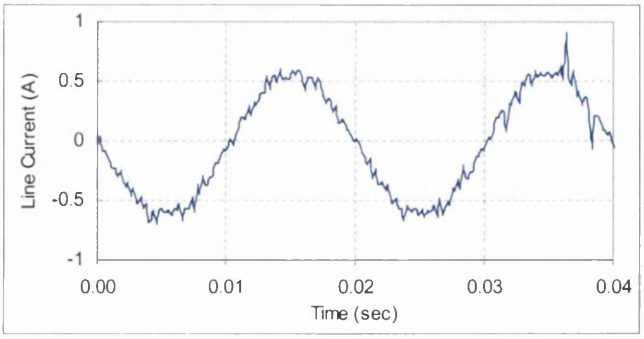
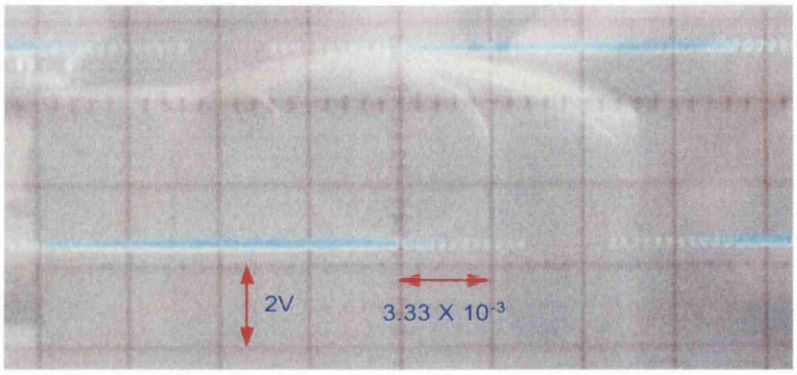
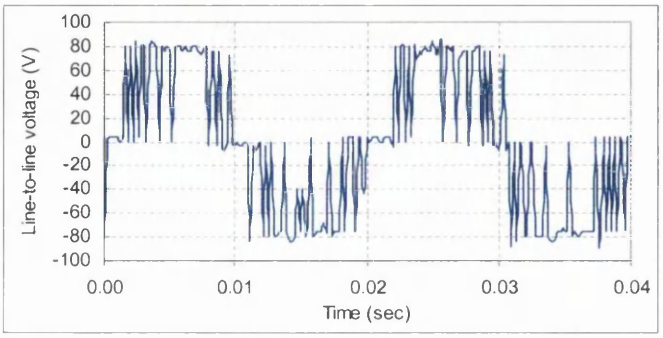


Figure 8-5: Experimental waveforms for the THI-SPWM at f_r =60; (a) PWM gating signal; (b) line-to-line voltage;(c) phase voltage; (d) line current.

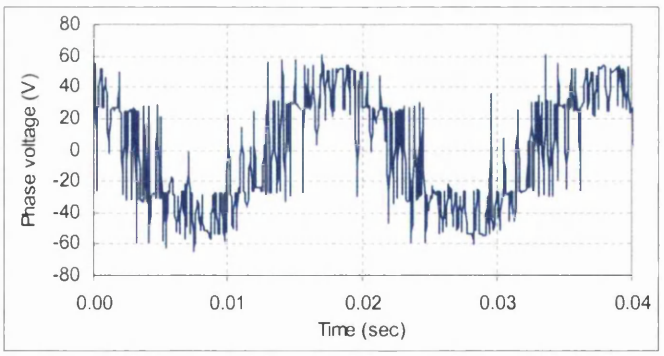
(a)



(b)



(c)



(d)

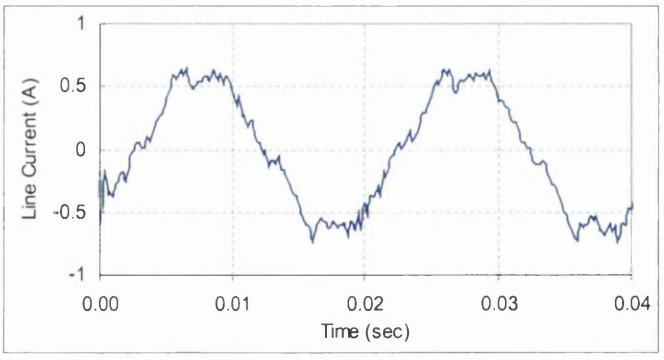


Figure 8-6: Experimental waveforms for the Modified-SPWM at f_r =60; (a) PWM gating signal; (b) line-to-line voltage;(c) phase voltage; (d) line current.

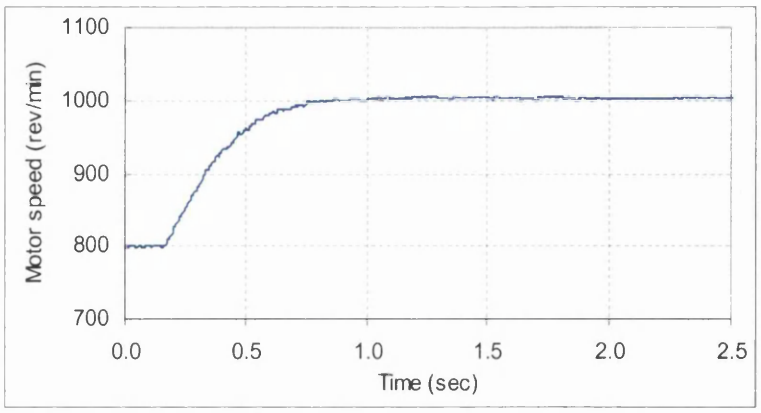
8.3 Characteristics of the induction motor under open-loop control test:

In this section, the transient dynamics of the induction motor are investigated under simple open-loop control test. This represents a very important step to initially verify the reliability of the designed program code before the induction motor is subjected to a full closed-loop operation. For this purpose, the conventional *Regular Asymmetric (RA)* SPWM technique is used in the present experimental test. The DC link voltage of the inverter is set to a 100V DC, while the switching frequency, f_s, is selected to be 900Hz, and the sampling time T_s is set to 5 msec. The speed of the induction motor is controlled by changing the value of the modulating frequency f_m, which is in turn changes the value of the supply voltage at a constant *V/f* ratio. The characteristics of some of the motor variables which portray the performance of the induction machine are measured in real-time and the corresponding data are stored in files, which can be accessed at any time.

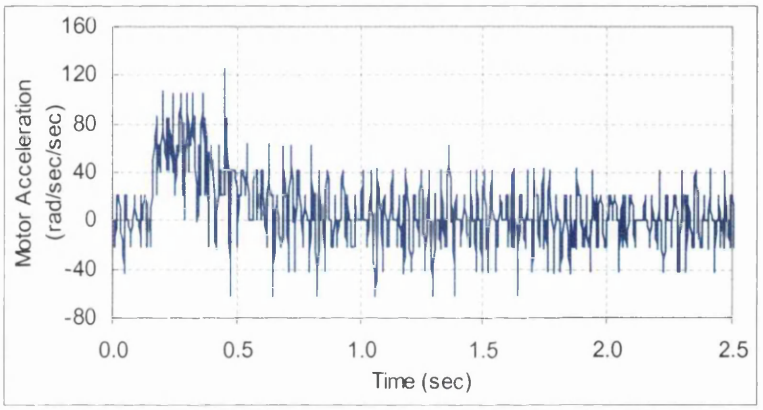
The experimental results of the open-loop test upon step change in speed command from 800 rev/min to 1000 rev/min, under no load, are shown in Figure 8-7. This figure depicts the profile of the motor speed response, the line current, and the motor acceleration. Inspection of these results indicates that the motor speed reaches the speed command in less than one second. On the other hand, the maximum transient line (stator) current reaches twice the steady state value. As the motor speed settles and reaches the steady state operation, the inrush current then converges to its steady state value after approximately 0.7 seconds. It should be emphasised, however, that small ripples are clearly noticed in the speed and acceleration curves. This is due to the effect of using a slightly high sampling time of 5 msec. In spite of this, the induction motor still exhibits reasonable transient dynamics with minimal steady state error.

In addition, Figure 8-8 demonstrates the simulation results obtained for the likewise test, under the same operating conditions. As shown, there is a fair agreement between the simulated and experimental results. In contrast, the transient shape of the motor line-to-line and phase voltages are almost similar to those obtained in the steady state except that the frequency of the generated pulses is varied. The experimental profile of the line-to-line voltage waveform and the phase voltage waveform in the steady state are depicted in Figure 8-9. Similarly, the simulated profiles of these waveforms are shown in Figure 8-10.

(a)



(b)



(c)

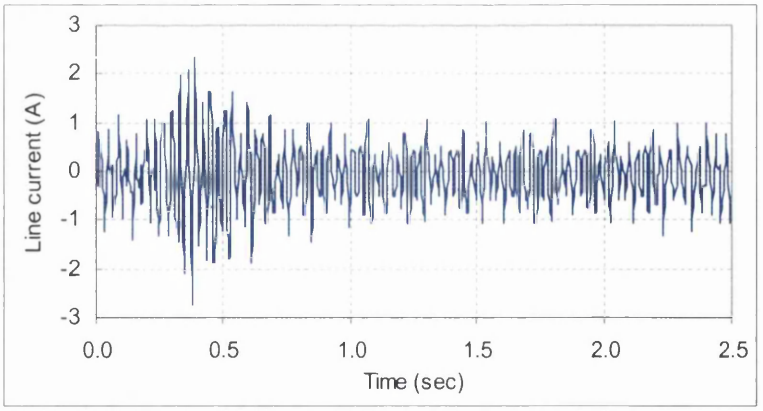
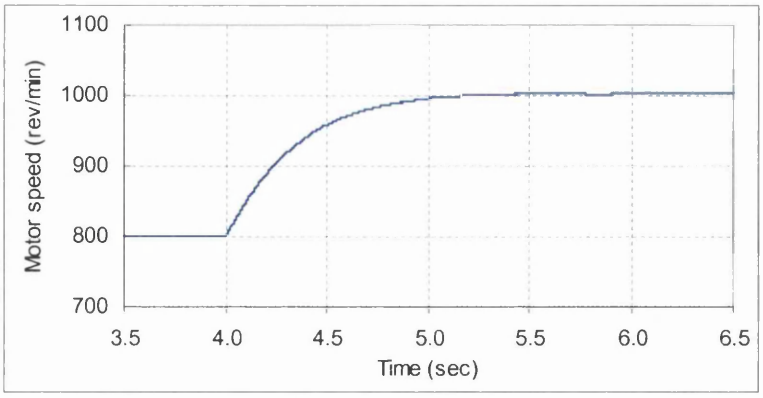
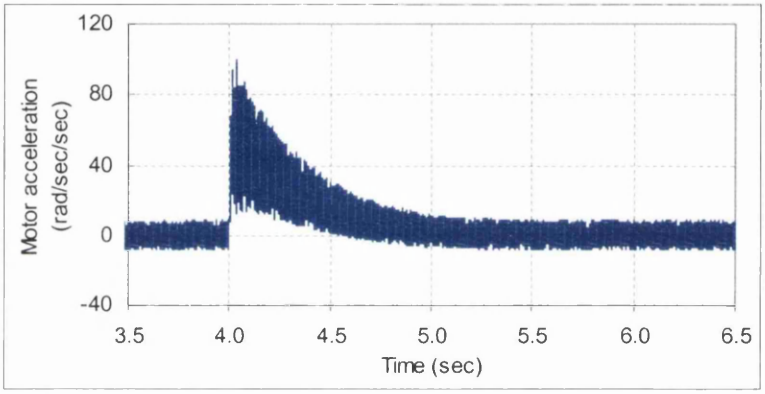


Figure 8-7: Experimental waveforms for the open-loop control test upon step change in the speed command; (a) motor speed; (b) motor acceleration; (c) line current.

(a)



(b)



(c)

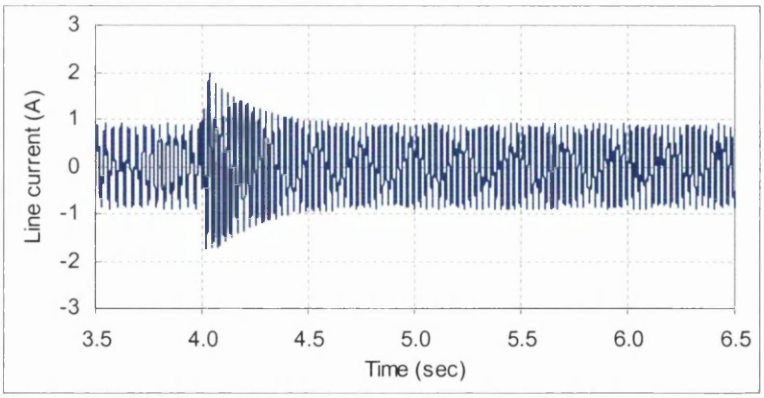
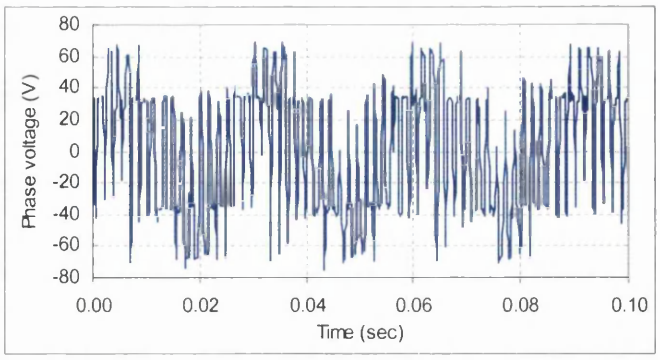


Figure 8-8: Simulation waveforms for the open-loop control test upon step change in the speed command; (a) motor speed; (b) motor acceleration; (c) line current.

(a)



(b)

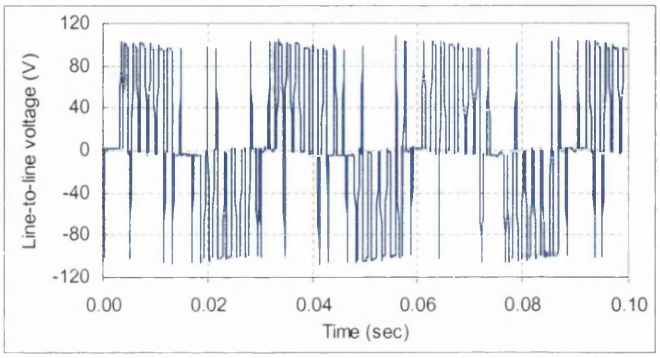
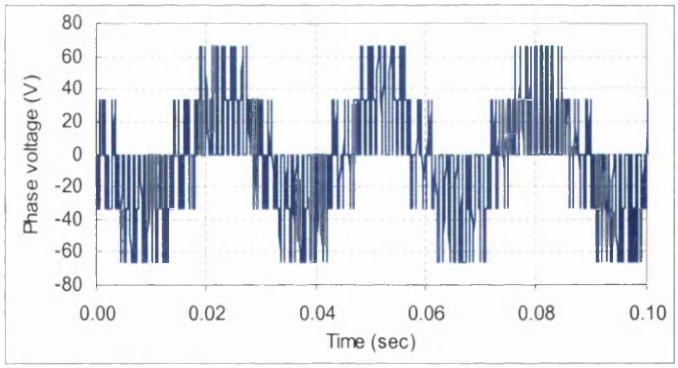


Figure 8-9: Experimental voltage waveforms for the open-loop control test in the steady state; (a) phase voltage; (b) line-to-line voltage.

(a)



(b)

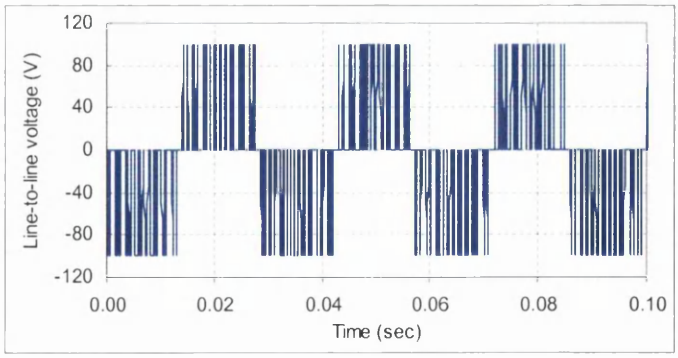


Figure 8-10: Simulated voltage waveforms for the open-loop control test in the steady state; (a) phase voltage; (b) line-to-line voltage.

8.4 Characteristics of the induction motor under closed-loop control test:

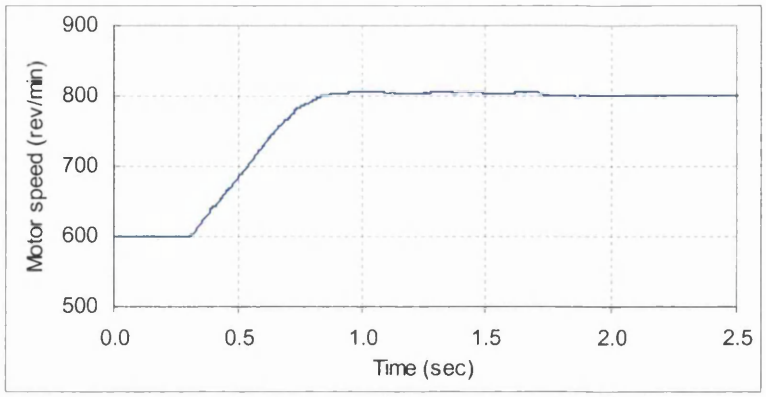
In this section, the assessment of the dynamic performance of the proposed speed drive system incorporating fuzzy logic control is considered. Several tests have been conducted in order to bring out the effectiveness of the designed control system upon step change in speed command and impact load disturbances. As in the open-loop control test, the conventional *Regular Asymmetric (RA)* SPWM technique is used in this test. The DC link voltage of the inverter is set to a 100V DC, while the switching frequency, f_s, is selected to be 900Hz, and the sampling time T_s is kept at 5 msec. In addition, the step response test of the fuzzy logic controller has been simplified by utilising a soft-start approach before actually commencing the closed-loop control operation. This is thoroughly explained in the preceding chapter.

8.4.1 Step change in speed command test:

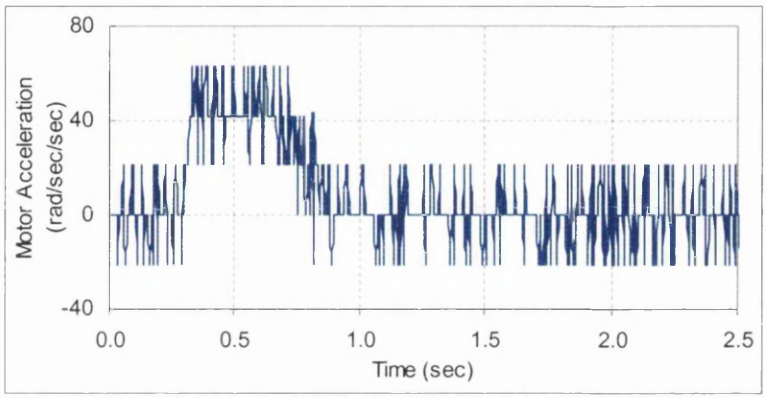
Throughout this test, the shaft of the induction motor has been de-coupled from that of the dc generator load in order to initially test the proposed closed-loop system at a low value of inertia. In other words the induction motor is operating in light running mode. Figure 8-11 and 8-12 show the dynamic responses of the induction motor due to a step change of 200 rev/min in the speed command applied when the induction motor was first running at 600 rev/min and 800 rev/min, respectively. Inspection of the results indicates that the fuzzy logic control follows the command speed reasonably well with a minimal steady state error. Also shown in both figures is that the speed command has been reached in about 0.6 seconds, with no overshoot.

Due to the increase in the slip frequency command (also shown in Figures 8-11 and 8-12) a torque is developed, which causes motor acceleration and thus the speed is building up. As shown, the slip frequency command is limited to a value of 35 rad/sec, at which the maximum operating torque is produced. Once the speed command is reached, the slip frequency then converges to a steady state value that is dictated by the value of the shaft friction constant. Moreover, the figures also illustrate the dynamic responses of the motor acceleration and the resulting line current for both tests. As shown, there is a quite high ripple in both current waveforms, but they don't show any severe effects in the speed response.

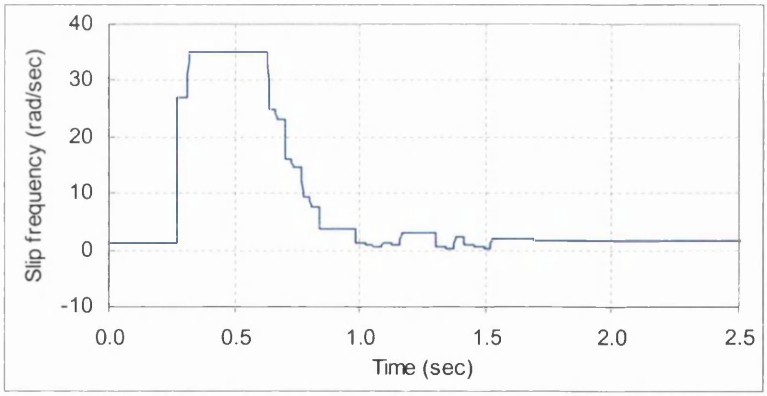
(a)



(b)



(c)



(d)

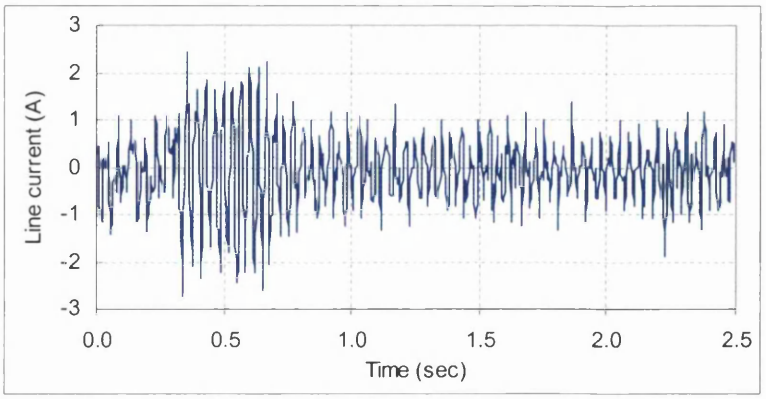
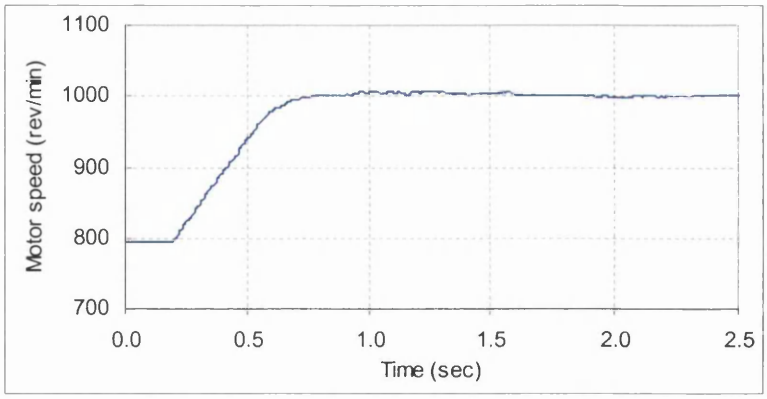
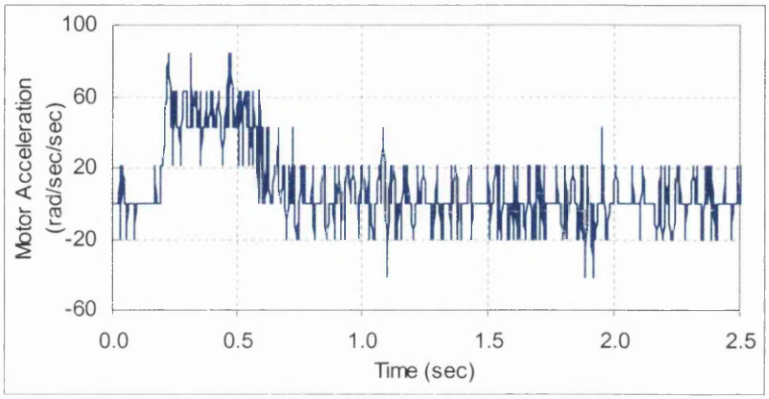


Figure 8-11: Experimental waveforms for the closed-loop control test upon step change in the speed command, from 600 rev/min to 800 rev/min; (a) motor speed; (b) motor acceleration; (c) slip frequency command; (d) line current.

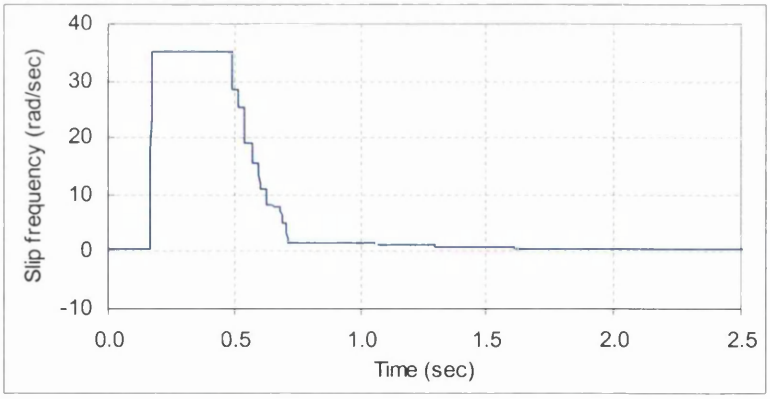
(a)



(b)



(c)



(d)

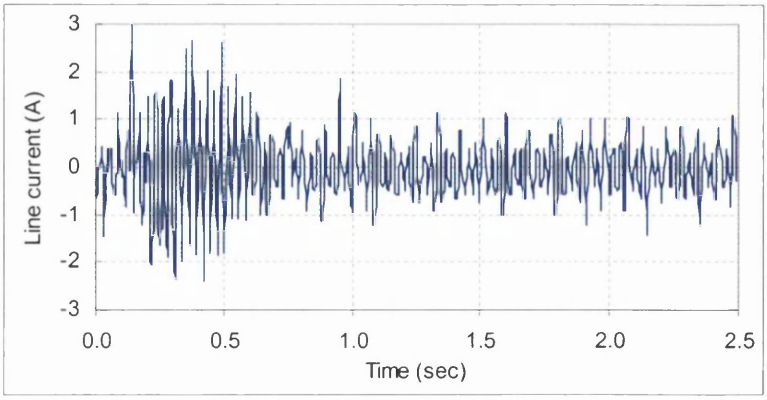


Figure 8-12: Experimental waveforms for the closed-loop control test upon step change in the speed command, from 800 rev/min to 1000 rev/min; (a) motor speed; (b) motor acceleration; (c) slip frequency command; (d) line current.

Figure 8-13 shows the dynamic response of the induction motor due to a step change in the speed command from 800 rev/min to 600 rev/min. This test verifies the robust operation of the proposed control system when the speed error becomes negative. Under this condition, the induction machine has a negative slip and operates as an induction generator, returning energy back to the inverter circuit. In other words, the induction machine is operating in the regeneration mode. It is clearly seen from this figure that due to the increase in the negative slip frequency command, a large negative torque is developed, which causes motor deceleration and thus the speed is decreasing.

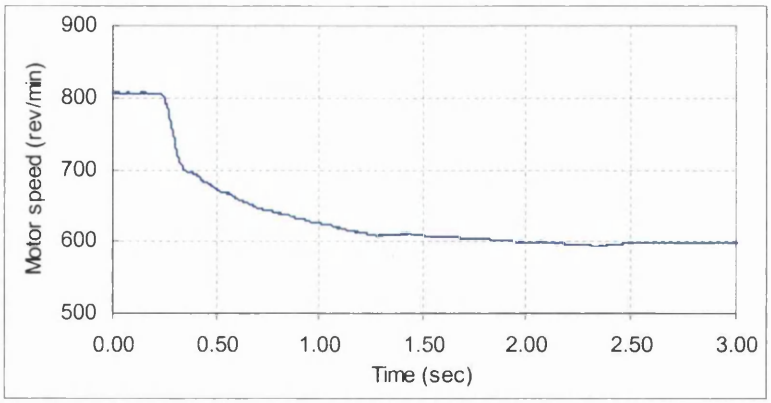
In addition, Figure 8-14 and 8-15 demonstrates the simulation results obtained for a step change in the speed command from 600 rev/min to 800 rev/min and from 800 rev/min to 600 rev/min, respectively. It should be noted that the same operating conditions applied in the experimental tests are considered here for the simulation. Inspection of these figures indicates that the experimental results (Figure 8-11 and 8-13) agree with those of the simulation regarding response time, speed error and slip frequency in the transient and steady state periods.

The differences, however, between simulation and experimental results concerning the dynamic response of the induction motor arise from the use of a lower switching time and an ideal acceleration measurement for the simulation. In spite all of this, the agreement between the simulation and experimental results is sufficient enough to prove the feasibility of the analysis of the proposed control system.

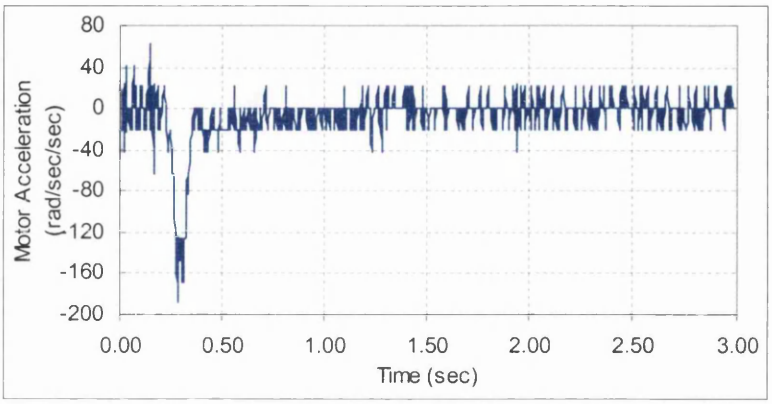
8.4.2 Invariance characteristic of the proposed control system:

In this section, the dynamic robustness of the designed fuzzy logic controller against parameters variation in the controlled system is examined. For this purpose, the shaft of the dc generator load is coupled to that of the induction motor in order to change the drive inertia and hence cause parameter variation. In this test, the DC link voltage of the inverter is extended to a 150V to provide sufficient torque capability in order to run higher drive inertia. Thus, the results obtained in this test would be comparable to the previous characteristic results of the induction motor as shown in Figure 8-11. Figure 8-16 demonstrates the practical results obtained for a step change in the speed command from 600 rev/min to 800 rev/min when the motor inertia has changed because of shaft coupling.

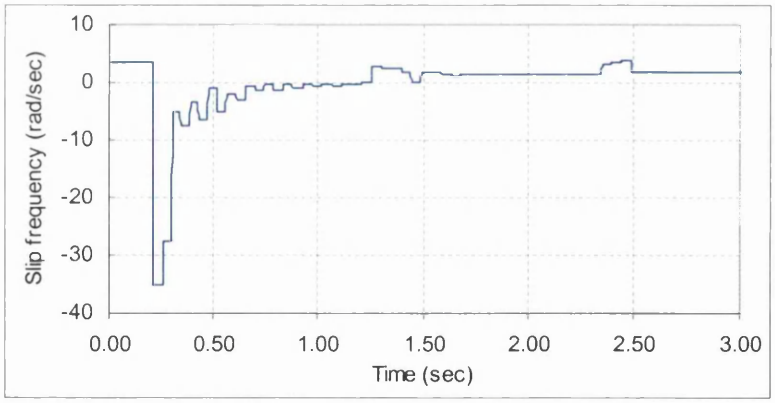
(a)



(b)



(c)



(d)

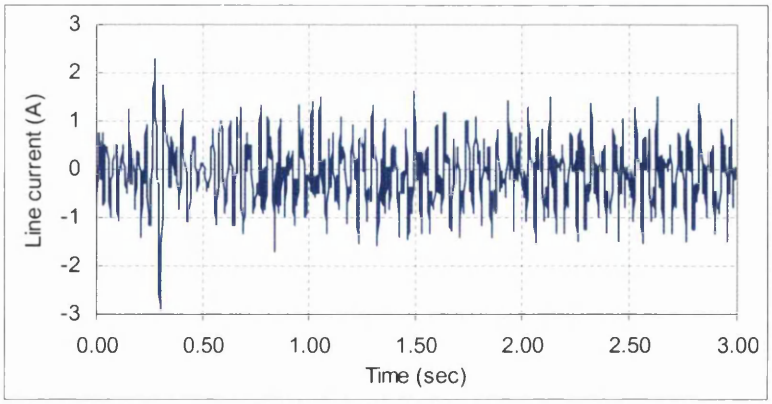
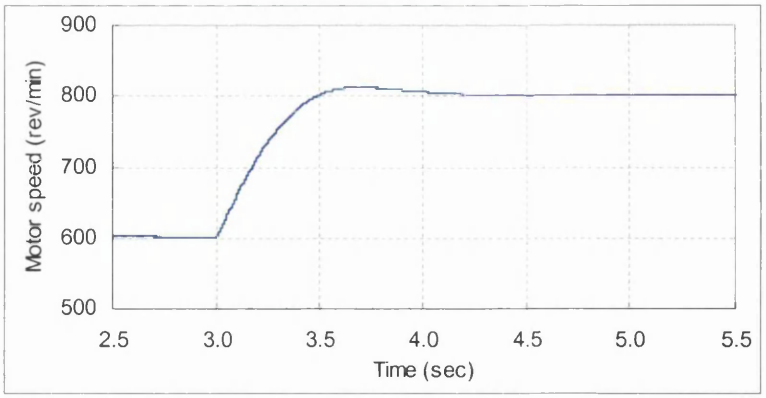
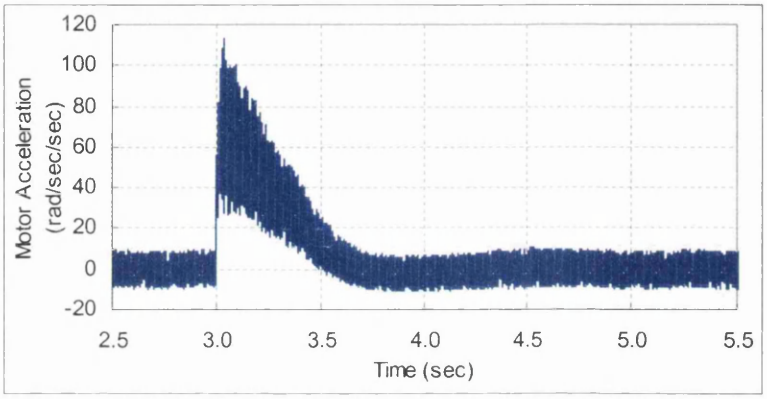


Figure 8-13: Experimental waveforms for the closed-loop control test upon step change in the speed command, from 800 rev/min to 600 rev/min; (a) motor speed; (b) motor acceleration; (c) slip frequency command; (d) line current.

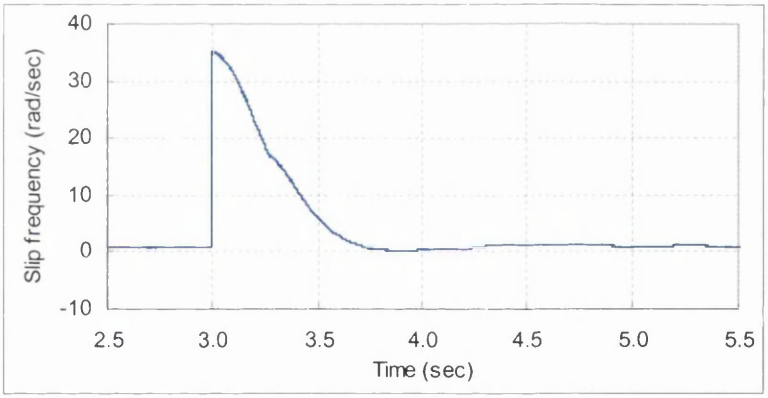
(a)



(b)



(c)



(d)

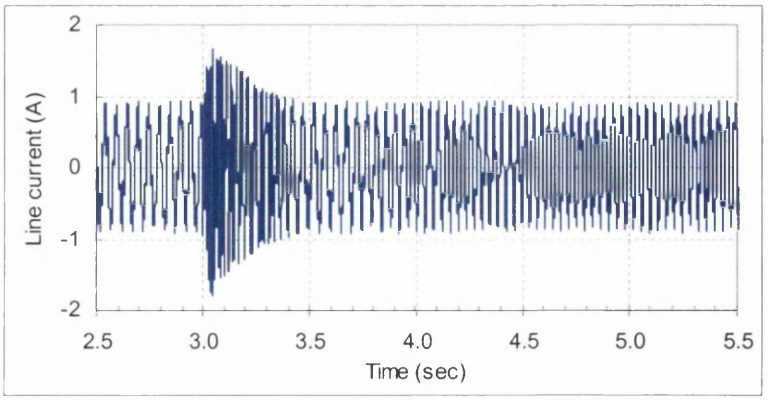
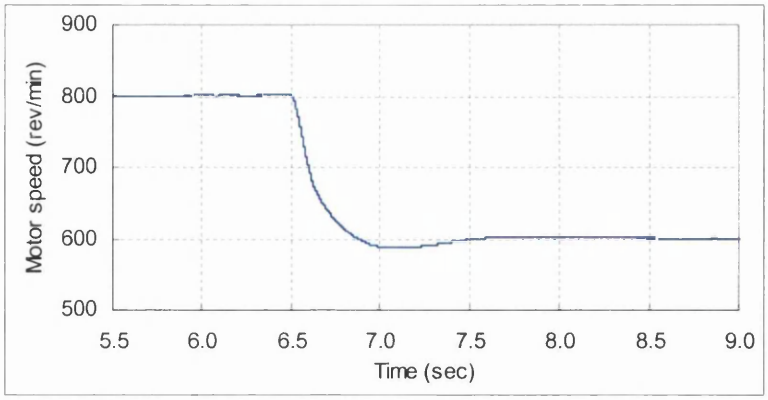
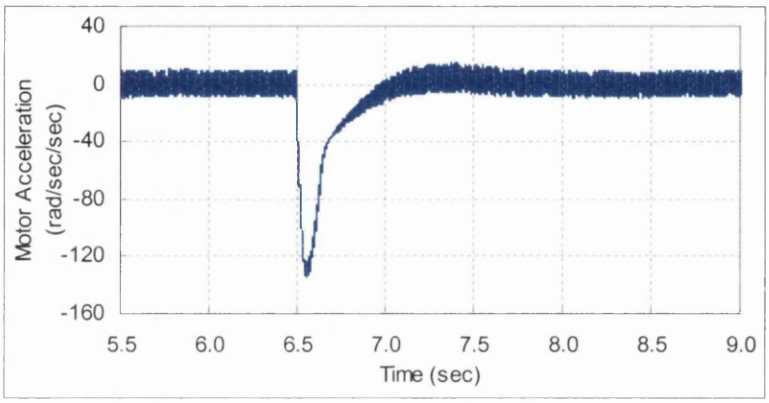


Figure 8-14: Simulation waveforms for the closed-loop control test upon step change in the speed command, from 600 rev/min to 800 rev/min; (a) motor speed; (b) motor acceleration; (c) slip frequency command; (d) line current.

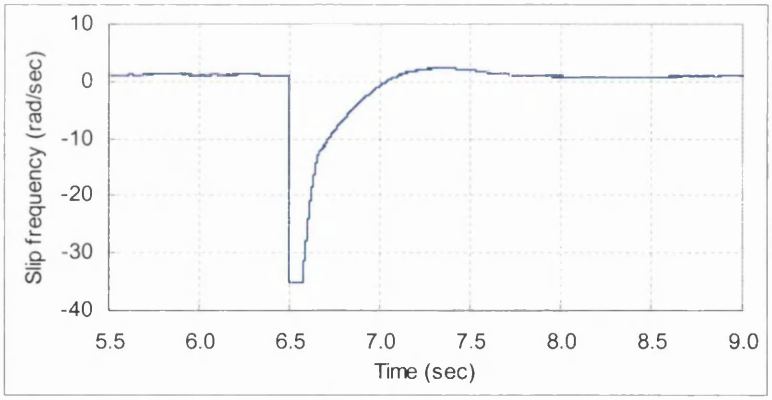
(a)



(b)



(c)



(d)

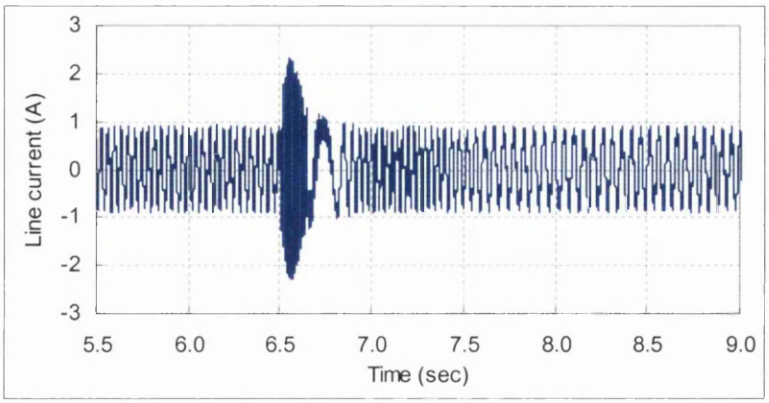


Figure 8-15: Simulation waveforms for the closed-loop control test upon step change in the speed command, from 800 rev/min to 600 rev/min; (a) motor speed; (b) motor acceleration; (c) slip frequency command; (d) line current.

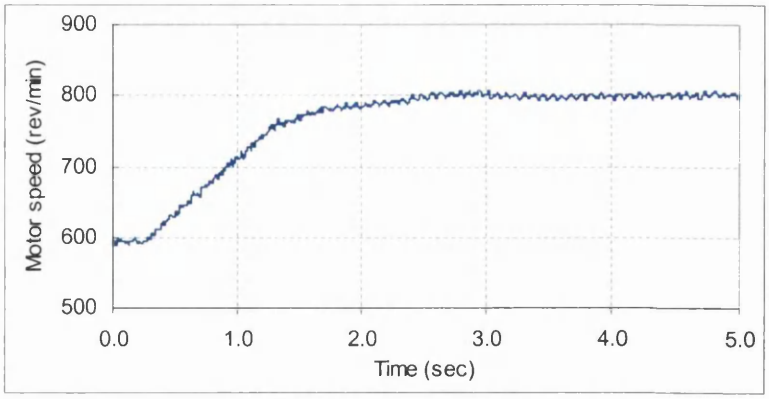
Inspection of this figure indicates that the fuzzy control system is insensitive to the change in the drive inertia and managed to provide a reasonable speed response, which is almost identical but slower to that shown in Figure 8-11. As seen in Figure 8-16, however, high ripple appeared in the speed and acceleration curves in both the transient and steady state periods. This is mainly due to the possible axis misalignment of the coupled shafts between the induction motor and the dc generator. Despite this, the fuzzy logic controller manages to produce a satisfactory dynamic performance. In addition, Figure 8-17 shows the simulation results obtained for the same test. It should be noted that the difference in the duration of the transient response in Figures 8-16 and 8-17 is due to the coupling with the dc generator that significantly increases the inertia of the drive.

8.4.3 Dynamic performance of the proposed control system during sudden load impact disturbances :

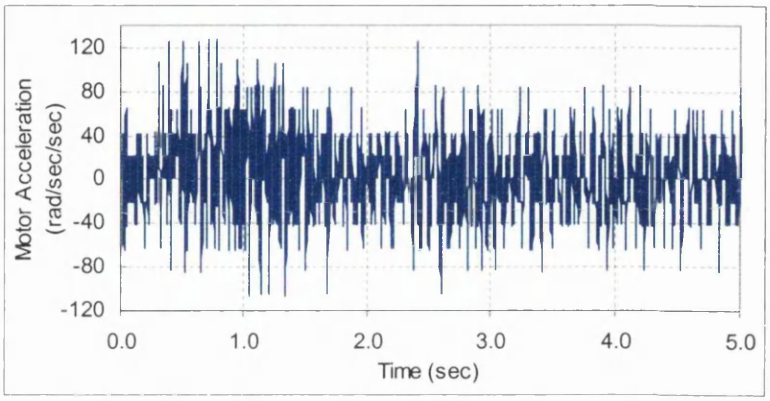
This section examines the disturbance rejection capabilities of the fuzzy controller when external load is suddenly applied to the motor shaft. The load disturbances are produced by varying the value of the loading resistor, which is connected to the armature terminals of the dc generator. Figures 8-18 demonstrates the load regulating characteristics of the induction motor upon step changes in load torque. Initially, the induction motor was running at 800 rev/min before load is applied to the shaft. As can be seen, the speed drop under impact load is less than 25 rev/min, and a recovery time is less than 1 sec, with minimal steady state error.

Moreover, the figure also shows the dynamic performance of the induction motor when the applied load is suddenly removed. Under this condition, the motor speed is swiftly increased due to the remaining high slip frequency command from the preceding loading effect. In spite of this the fuzzy controller quickly restores the speed to the command speed with a speed overshoot of less than 25 rev/min. In addition, Figure 8-20 shows the simulation results obtained for the same loading tests. These results show that good load regulating performances can be achieved with the proposed fuzzy logic controller.

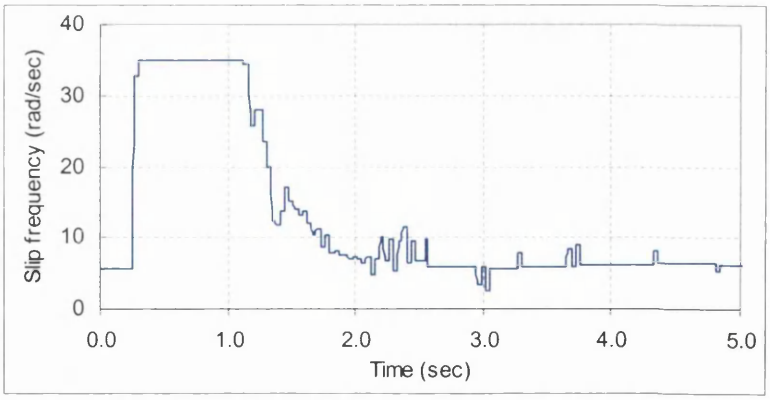
(a)



(b)



(c)



(d)

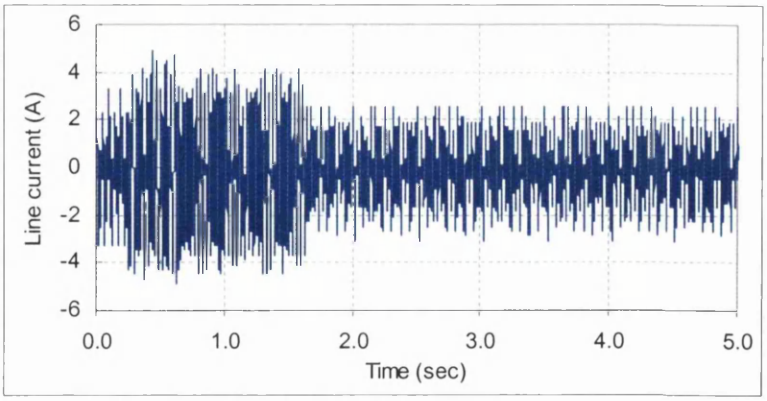
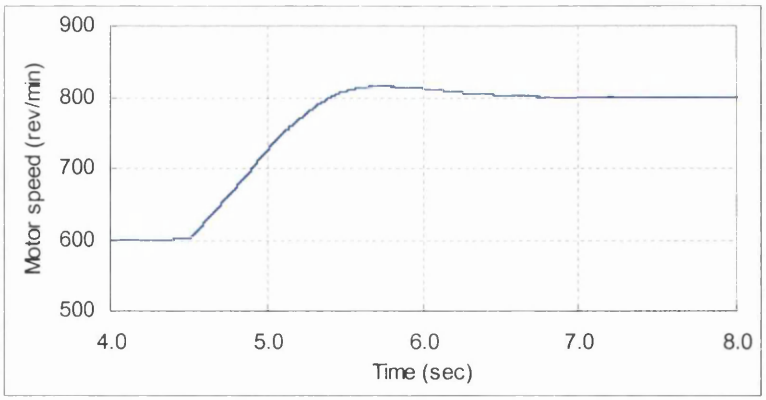
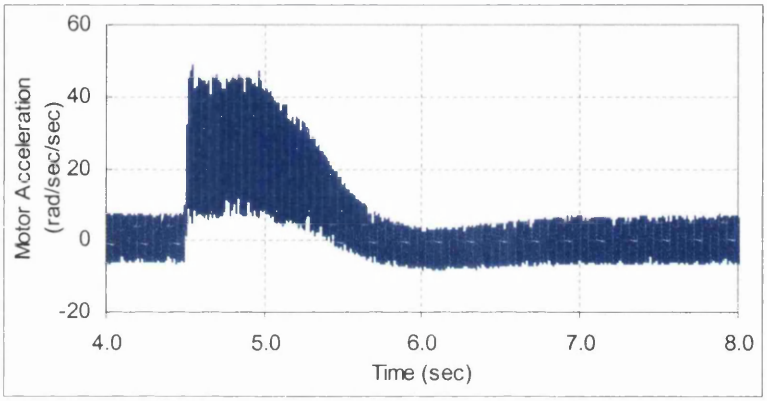


Figure 8-16: Experimental waveforms for the closed-loop control test upon step change in the speed command, from 600 rev/min to 800 rev/min, with induction motor coupled to dc generator; (a) motor speed;(b) motor acceleration; (c) slip frequency command; (d) line current.

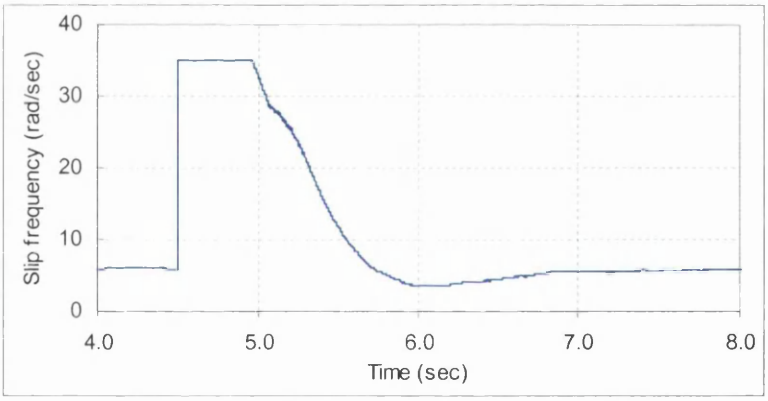
(a)



(b)



(c)



(d)

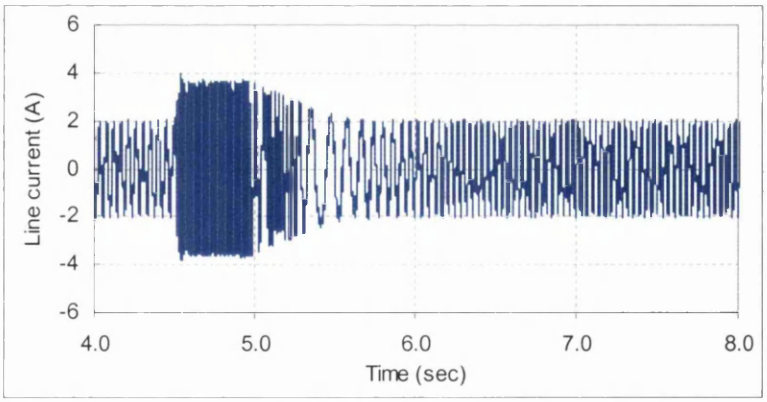
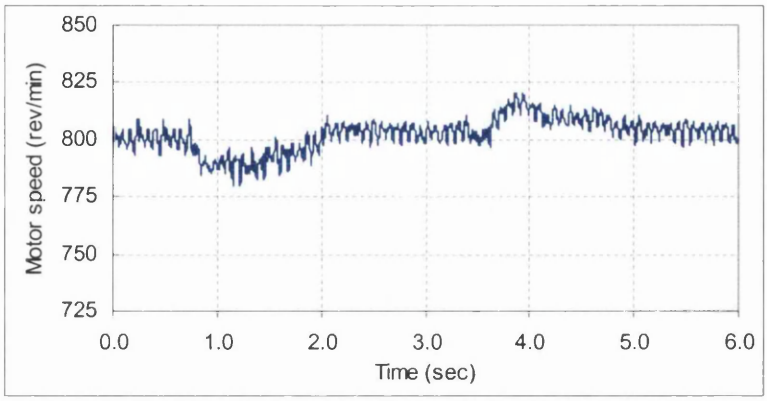
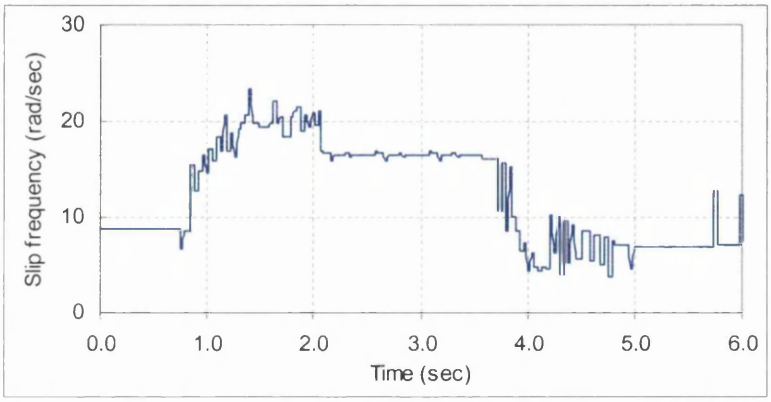


Figure 8-17: Simulation waveforms for the closed-loop control test upon step change in the speed command, from 600 rev/min to 800 rev/min, with induction motor coupled to dc generator; (a) motor speed;(b) motor acceleration; (c) slip frequency command; (d) line current.

(a)



(b)



(c)

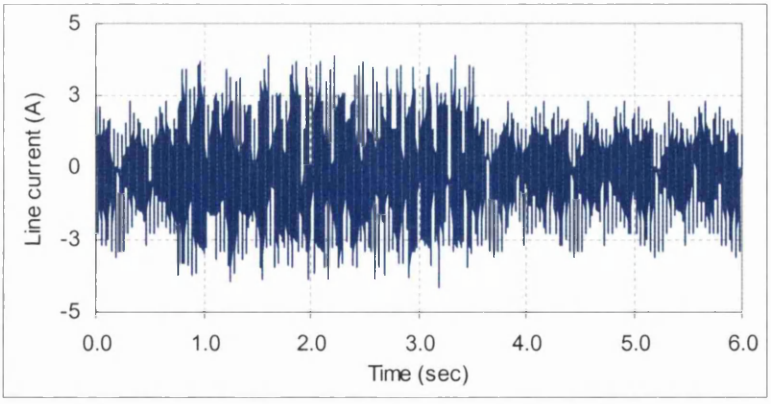
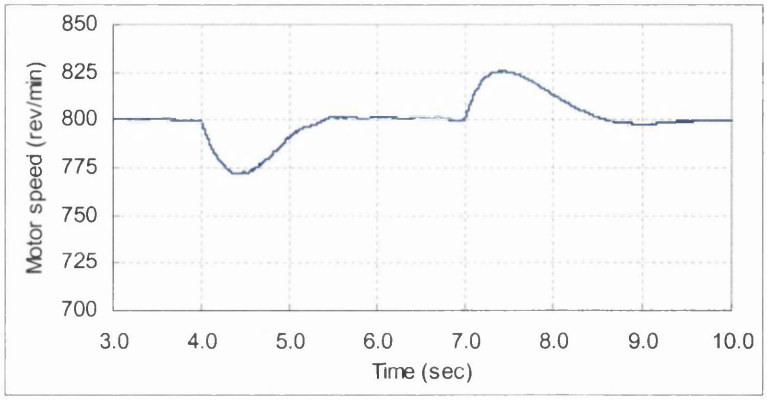
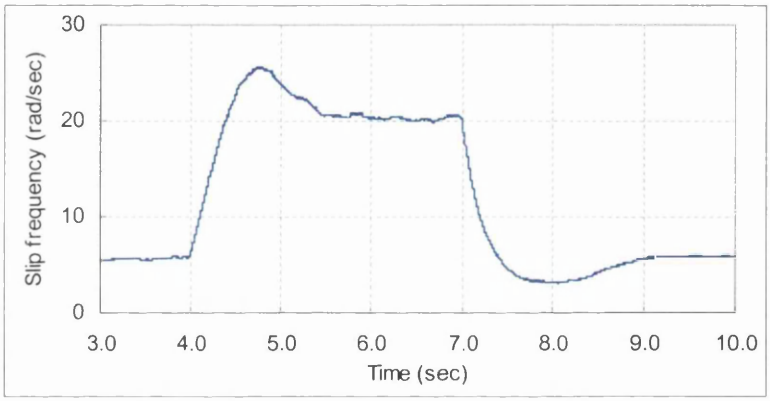


Figure 8-18: Experimental waveforms for the closed-loop control test upon step change in load torque ; (a) motor speed; (b) slip frequency command; (c) line current.

(a)



(b)



(c)

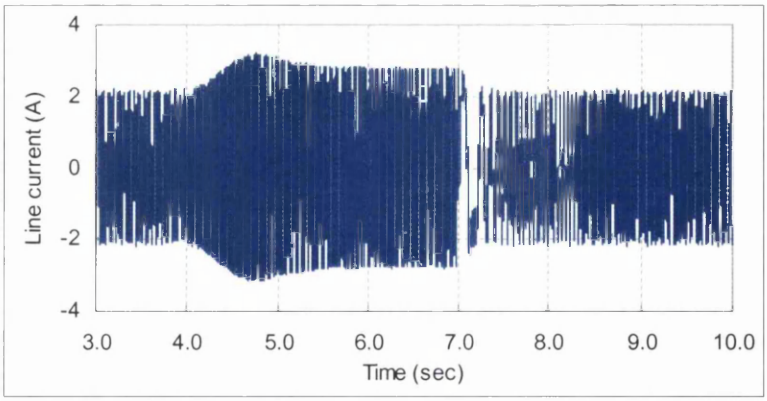


Figure 8-19: Simulation waveforms for the closed-loop control test upon step change in load torque; (a) motor speed; (b) slip frequency command; (c) line current.

8.5 Summary:

In this chapter, a brief investigation of the characteristics of some of the motor variables when the induction motor is subjected to different types of sinusoidal PWM signals for a modified switching pattern was presented. In addition, the performance of the induction motor under open-loop control test was shown. It is followed by illustrating the dynamic performance of the proposed speed control system, which demonstrated the validity of the designed FLC speed controller. Comparison studies between the acquired experimental results with those obtained via simulation, for the same operating conditions, were considered.

Chapter 9, Conclusions and recommendations:

9.1 Conclusions:

The improvement of ac drive performance, cost reduction, and the environmental need for energy efficiency are the essential driving forces for the growth of modern ac speed drives. Different control techniques of varying degrees of complexity have made the application of induction motors feasible in high performance speed drive applications where only dc motors were previously available. The choice of a suitable control technique depends on the nature of the control application.

In various existing ac drive systems, **scalar control** (*V/f*) and **vector control** (also known as field oriented control) are the most widely accepted control techniques. However, the *V/f* ac drives are still widely accepted as being less complex than its counterpart vector ac drives, of lower cost and easier to set-up. In addition, the *V/f* ac drives still retain a strong presence in many industrial applications for a stand alone operation and also for certain applications requiring the control of multiple ac motors on a single drive.

In many speed drive applications which incorporates either *V/f* or vector control, the prime objectives of any speed controller are to be insensitive to system parameters variation, has a rapid tracking performance of command speed changes without overshoot, and being robust against external load disturbances. Therefore, the aim of this research work is to design a relatively simple, cost effective and robust speed controller for an ac induction motor (squirrel cage type) drive.

It should be stressed that the designed controller ought to be robust in terms of speed tracking, load disturbance rejection, and parameter variations without the requirement for complex control algorithms. For this purpose, comprehensive investigations of state-of-the-art modern control systems, which include fuzzy logic and sliding mode control have been discussed. In addition, the application of fuzzy logic concepts to reduce the chattering phenomenon typically inherited in the sliding mode control has been successfully presented in Chapter 5. The new integrated control algorithm was designated as '*fuzzy sliding mode control*' (FSLMC).

The main principles underlying fuzzy logic and sliding mode control schemes have been reviewed in Chapters 4 and 5, respectively. The basic theory and general mathematical representation of both controllers have been also presented along with a brief review of their recent applications in ac speed drive systems. Moreover, the development of the new FSLMC control scheme that is based on the concept of "*reaching law*" has been thoroughly discussed.

The developed algorithms of the three preceding speed controllers have been initially assessed through computer modeling and simulation using MATLAB and SIMULINK. The step was found useful to initially design and select sufficient speed controller for the closed loop speed control of an induction motor, in which the design can be tested without costly and time-consuming experiments with hardware. For a comprehensive transient study of the drive system, the simulation model of the induction motor has been created using a set of dynamical equations which were formulated in the synchronous reference frame (two-axis model) as given in Chapter 3.

It has been shown through an in-depth simulation study given in Chapter 6, that the proposed fuzzy logic control (FLC) scheme attained a good transient performance for the speed drive system in comparison to the conventional sliding mode control and the new integrated FSLMC. Furthermore, the design simplicity of the FLC system has made it virtually attractive for the ease of practical implementation of the proposed drive system. The assessments of these controllers were based on their capabilities of providing good speed tracking performance, and robustness against external load disturbances and parameter variations.

The digital implementation of the proposed FLC algorithms has been done on a single chip, Intel 80C196KC 16-bit embedded microcontroller, a low cost derivative of the MCS-96 architecture. Some modifications have been made to simplify the designed fuzzy logic algorithms in order to ease its real time implementation on the microcontroller. One of these modifications is that the input and output variables of the FLC system have been discretised into thirteen quantised levels, with which a look-up table has been constructed off-line using the design procedures described in Chapter 4. This look-up table was then stored in the external RAM of the microcontroller, which is evaluated during the run-time. The implementation of the proposed speed drive system includes additional tasks such as speed measurement, digital filtering, calculation and generation of PWM pulses. All of these tasks have been easily implemented on the existing processor, resulting in a cost effective and improved reliability speed drive.

Extensive practical tests of the proposed variable speed drive system have been carried out to verify the validity of the simulation analysis of the proposed FLC system. Several tests have been conducted in order to bring out the effectiveness of the designed control system upon step change in speed command and impact load disturbances. The insensitivity property of the proposed control scheme against parameter variations has been also presented.

The obtained experimental results have demonstrated the feasibility of using a simple fuzzy logic control scheme in the induction motor speed control system. In addition, the practical performance of the induction motor speed drive system has been found to be in good agreements with the simulations and theoretical analysis.

9.2 Recommendations for future investigation:

The prime goals of the proposed induction motor speed drive system have been fulfilled to a great degree of achieving sufficient dynamic performances. The accomplished goals have been demonstrated through extensive series of simulation and experimental studies. To further improve the presently proposed speed control system, few suggestions are proposed as follows:

- In theory, fuzzy controller is regarded as a nonmathematical control system in comparison to the conventional feedback control algorithms such as the sliding mode control and the classical PID control. Although they have been successfully used in many high performance applications such as speed drive systems, their theoretical analysis such as stability and robustness have not been very well developed (Lo and Kuo 1998; Yi and Chung 1998). Therefore, it is necessary to provide systematic approaches to characterise the fuzzy logic algorithms in such a way that stability and robustness are ensured in the fuzzy logic controller.

- In the real time implementation of fuzzy logic controller, it is beneficial to include scaling factors to the input and output variables of the controller, which can be tuned for a desired performance. Typically, the input scaling factor has a great influence on the sensitivity of the fuzzy logic controller, while the output scaling factor has a great influence on the stability of the control system. The tuning of these scaling factors to obtain their appropriate values is usually done via extensive simulations tests.

REFERENCES

Abbondanti, A., and Brennen, M. B., (1975), "Variable speed induction motor drives use electronic slip calculator based on motor voltages and currents", IEEE Trans. Indus. App., Vol. IA-11, No. 5, pp. 483-488.

Abed, S. K., (1995), "An Investigation into Real Time Controlled Three-Phase Induction Motor Drive System". Ph.D. Thesis, University of Wales Swansea.

Addoweesh K. E., Shepherd W., and Hulley L. N., (1989), "Induction motor speed control using a microprocessor-based PWM inverter", IEEE Trans. Ind. Electron., Vol. 36, No. 4, pp. 516-522.

Afonso, J. L., Fonseca, J., Martins J. S., and Couto, C. A., (1997), "Fuzzy logic techniques applied to the control of a three-phase induction motor", Proc. of the IEEE International Symposium on Industrial Electronics (ISIE), Gumaraes, Portugal, pp. 1179-1184.

Aminzadeh, F., and Jamshidi, M., (1994), "Soft computing: fuzzy logic, neural network, and distributed artificial intelligence", PTR Prentice-Hall, Inc.

Anagreh, Y., (1997), "Real-time sliding mode control of a brushless dc motors". Ph.D. Thesis, University of Wales Swansea.

Athani, V. V., and Deshpande, S. M., (1980), "Microprocessor control of a three-phase inverter in induction motor speed control system", IEEE Trans. Indus. Elect. and Control Instrumentation, Vol. IECI-27, No. 4, pp. 291-298.

Baghali, L., Razik, H., and Rezzoug, A., (1997), "Neuro-fuzzy controller in a field oriented control for induction motors", 7th European Conference on Power Electronics and Applications (EPE), Tandheim, Norway, pp. 1.096-1.101.

Baghli, L., Razik, H., and Rezzoug, A., (1997), "Comparisons between fuzzy and classical speed control within a field oriented method for induction motors", 7th European Conference on Power Electronics and Applications (EPE), Tandheim, Norway, pp. 2.444-2.448.

Bellini A., Figalli G., and Ulivi G., (1985), "A high-performance microprocessor-based control circuit for variable frequency inverter", IEEE Trans. Ind. Electron., Vol. IE-32, No. 1, pp. 62-70.

Benchaib, A., Rachid, A., and Audrezet, E., (1999), "Sliding mode input-output linearisation and field orientation for real-time control of induction motors", IEEE Trans. Power Electronics, Vol. 14, No. 1, pp. 3-13.

Bird, I. G., and De La Parra, H. Z., (1997), "DSP implementation of a fuzzy based direct flux and torque controller", 7th European Conference on Power Electronics and Applications (EPE), Tandheim, Norway, pp. 2.415-2.420.

Birou, I., Imecs, M., Kekemen, A, and Eutebach, T., (1996), "Implementation of fuzzy control for induction motor drive using MC68000 processor", Proc. of International Power Conversion and Intelligent Motion Conference (PCIM), Nurenberg, Germany, pp. 225-235.

Bojadziev, G., and Bojadziev, M., (1995), "Fuzzy sets, fuzzy logic, applications", World Scientific Pub. Co. Pte. Ltd.

Bolognani, S., and Zigliotto, M., (1996), "Fuzzy logic control of a switched reluctance motor drive", IEEE Trans. Indus. App., Vol. 32, No. 5, pp. 1063-1068.

Bonissone, P. P, Badami, V., Chiang, K. H., Khefkar, P. S., Marcelle, K. W., and Schutten, M. J, (1995), "Industrial applications of fuzzy logic at general electric", Proceedings of the IEEE, Engineering App. of Fuzzy Logic, pp. 450-465.

Boost M. A., and Ziogas P. D., (1988), "State-of- the-art carrier PWM techniques: a critical evaluation", IEEE Trans. Ind. Appl., Vol. 24, No. 2, pp. 271-280.

Bose, B. K., (1982), "Adjustable speed ac drives-a technology status review", Proc. of IEEE, Vol. 70, No. 2, pp. 116-135.

Bose, B. K., (1988), "Technology trends in microcomputer control of electrical machines", IEEE Trans. Ind. Electron., Vol. 35, No. 1, pp. 160-177.

Bowes S. R., (1975), "New sinusoidal pulse width-modulated inverter", IEE Proc., Vol. 122, No. 11, pp. 1279-1285.

Bowes S. R., and Clements R. R., (1982), "Computer-aided design of PWM inverter systems", IEE Proc., Vol. 129, Pt. B, No. 1, pp. 1-17.

Bowes S. R., and Midoun A., (1988), "Microprocessor implementation of new optimal PWM switching strategies", IEE Proc., Vol. 135, Pt. B, No. 5, pp. 269-279.

Bowes S. R., and Mount M. J., "Microprocessor control of PWM inverters", (1981), IEE Proc., Vol. 128, Pt. B, No. 6, pp. 293-305.

Brian, A., and Breiner, M., (1995), "MATLAB for Engineers", Addison-Wesley Publishing Co.

Brown, R. G., "Smoothing, forecasting and prediction of discrete time series", Prentice-hall international, Inc., 1963.

Cerruto, E., Consoli, A, Raciti, A., and Testa, A., (1997), "Fuzzy adaptive vector control of induction motor drives", IEEE Trans. Power Electronics, Vol. 12, No. 6, pp. 1028-1040.

Chalam, V. V., (1987), "Adaptive control systems, techniques and applications", Marcel Dekker Inc.

Chalia, M. B., and Alouani, A. T., (1995), "Sliding mode control synthesis using fuzzy logic", Proc. Amer. Cont. Conf., Seattle, WA, pp. 1528-1532.

Chang, F., Liao, H., and Chang, S., (1990), "Position control of dc motors via variable structure systems control: a chattering alleviation approach", IEEE Trans. Indus. Elect., Vol. 37, No. 6, pp. 452-459.

Chang, S. L., and Zadeh, L. A., (1972), "On fuzzy mapping and control", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-2, No. 1, pp. 30-34.

Chattopadhyay, A. K., (1976), "Digital computer simulation of an adjustable-speed induction motor drive with a cycloconverter-type thyristor-commutator in the rotor", IEEE Trans. Ind. Electron. and Control Inst., Vol. IECI-23, No. 1, pp. 86-91.

Cheng, F., and Yeh, S., (1993), "Application of fuzzy logic in the speed control of ac servo system and intelligent inverter", IEEE Trans. Energy Conversion, Vol. 8, No. 2, pp. 312-318.

Chung, S. K., Lee, J. H., Ko, J. S., and Youn, M. J., (1995), "Robust speed control of brushless direct-drive motor using integral variable structure control", IEE Proc.-Electr. Power App., Vol. 142, No. 6, pp. 361-370.

Cunningham, E. P., (1992), "Digital filtering: an introduction", Houghton Mifflin Company.

Da Silva, W. G., and Acarnley, P. P., (1997), "Fuzzy logic controlled dc motor drive in the presence of load disturbance", 7th European Conference on Power Electronics and Applications (EPE), Tandheim, Norway, pp. 2.386-391.

Dubois, D., and Prade, H., (1980), "Fuzzy sets and systems: theory and applications", Academic Press Inc.

Esfandiari, F., and Khalil, H. K., (1991), "Stability analysis of a continuous implementation of variable structure control", IEEE Trans. Autom. Cont., Vol. 36, No. 5, pp. 616-619.

Ficarra, M. C., Eguilaz, J. M., and Peracaula, J., (1996), "Fuzzy control of an induction motor with compensation of system dead-time", Proc. of the IEEE Power Electronics Specialist Conference (PESC), Baveno, Italy, pp. 677-681.

Fodor, D., Katona, Z., and Vass, J., (1996), "On fuzzy logic speed control for vector controlled ac motors", 4th International Workshop on Advanced Motion Control (AMC), MIE University, TSU-City, Japan, pp. 186-191.

Gabriel, R., Leonhard, W., and Nordby, C. J., (1980), "Field oriented control of a standard ac motor using microprocessor ", IEEE Trans. Indus. App., Vol. IA-16, No. 2, pp. 186-192.

Gao, W., and Hung, J. C., (1993), "Variable structure control of nonlinear systems: a new approach", IEEE Trans. Indus. Elect., Vol. 40, No. 1, pp. 45-55.

Garces, L. J., (1980), "Parameter adaptation for the speed-controlled static ac drive with a squirrel-cage induction motor", IEEE Trans. Indus. App., Vol. IA-16, No. 2, pp. 173-178.

Gastli, A., and Matsui, N., (1992), "Stator flux controlled V/f PWM inverter with identification of IM parameters", IEEE Trans. Indus. Elect., Vol. 39, No. 4, pp. 334-340.

Ghani, S. N., (1988), "Digital computer simulation of three-phase induction machine dynamics-A generalized approach", IEEE Trans. Ind. Appl., Vol. 24, No. 1, pp. 106-113.

Guangdong, H., Jiang, J. P., Shen, J., and Lei, J., (1997), " A fuzzy variable structure control scheme for AC servo drive system", Proceedings of the IEEE International Conference on Electric Machines and Drives, Milwaukee, WI, pp. MC3-3.1-MC3-3.3.

Guillemin, P., (1996), "Fuzzy logic applied to motor control", IEEE Trans. Industry Applications, Vol. 32, No. 1, pp. 51-56.

Guo, Y., (1994), "A real time speed control system for DC drives", M.Phil. thesis, University of Wales Swansea.

Gupta, T., Boudreaux, R. R., Nelms, R. M., and Hung, J. Y., (1997), "Implementation of a fuzzy controller for DC-DC converters using an inexpensive 8-b micro-controller", IEEE Trans. Industrial Electronics, Vol. 44, No. 5, pp. 661-669.

Harashima, F., Hashimoto, H., and Kondo, S., (1985), "Mosfet converter-fed position servo system with sliding mode control", IEEE Trans. Indus. Elect., Vol. IE-32, No. 3, pp. 238-244.

Hava A. M., Kerkman R. J., and Lipo T. A., (1999), "Simple analytical and graphical methods for carrier-based PWM-VSI drives", IEEE Trans. Power Electron., Vol. 14, No. 1, pp. 49-61.

Heber, B., Xu, L., and Tang, Y., (1997), "Fuzzy logic enhanced speed control of an indirect field oriented induction machine drive", IEEE Power Electronics, Vol. 12, No. 5, pp. 772-778.

Hilloowala, R. M., and Sharaf, A. M., (1992), "A rule-based fuzzy logic controller for a PWM inverter in photo-voltaic energy conversion scheme", IEEE Technology Update Series; Fuzzy Logic Technology and Application, Chapter 6, pp. 304-311.

Ho, E. Y. Y., and Sen, P. C., (1986), "Digital simulation of PWM induction motor drives for transient and steady-state performance", IEEE Trans. Ind. Electron., Vol. IE-33, No. 1, pp. 66-77.

Ho, E. Y. Y., and Sen, P. C., (1990), "A microcontroller-based induction motor drive system using variable structure strategy with decoupling", IEEE Trans. Indus. Elect., Vol. 37, No. 3, pp. 227-235.

Ho, E. Y. Y., and Sen, P. C., (1991), "Control dynamics of speed drive systems using sliding mode controllers with integral compensation", IEEE Trans. Indus. App., Vol. 27, No. 5, pp. 883-235-892.

Hubert, C. I., (1991), "Electric machines theory, operation, applications, adjustment, and control", Macmillan Publishing Co.

Hung, J. Y., Gao, W., and Hung, J. C., (1993), "Variable structure control: a survey", IEEE Trans. Indus. Elect., Vol. 40, No. 1, pp. 2-21.

Hung, J. Y., Nelms, R. M., and Stevenson, P. B., (1994), "An output feedback sliding mode speed regulator for dc drives", IEEE Trans. Indus. App., Vol. 30, No. 3, pp. 691-698.

Ibrahim, Z., and Levi, E., (1998), "A detailed comparative analysis of fuzzy logic and PI speed control in high performance drives", Proc. IEE International Power Electronics and Variable Speed Drives Conference (PEVD), London, UK, Conference Publications No. 456, pp. 638-643.

Isaka, S., and Sebald, A. V., (1992), "An optimisation approach for fuzzy controller design", IEEE Trans. Systems, Man and Cybernetics, Vol. 22, No. 6, pp. 1469-1473.

Ishigame, A., Furukawa, T., Kawamoto, S., and Taniguchi, T., (1993), "Sliding mode controller design based on fuzzy inference for nonlinear systems", IEEE Trans. Indus. Elect., Vol. 40, No. 1, pp. 64-70.

Jang, J. R, and Sun, C-T., (1995), "Neuro-fuzzy modeling and control", Proceedings of the IEEE, Engineering App. of Fuzzy Logic, pp. 378-406.

Jones, C. V., (1967), "The unified theory of electrical machines", Butterworth and Co. Ltd.

Jordan, H. E., (1965), "Analysis of induction machines in dynamic systems", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-84, No. 11, pp. 1080-1088.

Kawamura, A., Itoh, H., and Sakamoto, K., (1994), "Chattering reduction of disturbance observer based sliding mode control", IEEE Trans. Indus. App., Vol. 30, No. 2, pp. 456-461.

Keiser, B. E., (1981), "Principles of electromagnetic compatibility", Artech House, Inc.

Koga, K., Ueda, R., and Sonoda, T., (1992), "Constitution of V/f control for reducing the steady-state speed error to zero in induction motor drive system", IEEE Trans. Indus. App., Vol. 28, No. 2, pp. 463-471.

Krause, P. C., and Lipo, T. A., (1969), "Analysis and simplified representations of a rectifier-inverter induction motor drive", IEEE Trans. Power App. and Systems, Vol. PAS-88, No. 5, pp. 588-596.

Krause, P. C., and Thomas, C. H., (1965), "Simulation of symmetrical induction machinery", IEEE Trans. Power App. and Systems, Vol. PAS-84, No. 11, pp. 1038-1053.

Krause, R., Gebhardt J., and Klawonn, F., (1994), "Foundations of fuzzy systems", John wiley and sons.

Kubo, K., Watanabe, M., Ohmae, T., and Kamiyama, K., (1985), "A fully digitalised speed regulator using multi-microprocessor system for induction motor drives", IEEE Trans. Indus. App., Vol. IA-21, No. 4, pp. 1001-1007.

Kung, Y., and Liaw, C., (1994), "A fuzzy controller improving a linear model following controller for motor drives", IEEE Trans. Fuzzy Systems, Vol. 2, No. 3, pp. 194-202.

Lai, M., Nakano, M., and Hsieh, G., (1996), "Application of fuzzy logic in the phase-locked loop speed control of induction motor drive", IEEE Trans. Industrial Electronics, Vol. 43, No. 6, pp. 630-639.

Lee, C. C., (1990a), "Fuzzy logic in control systems: fuzzy logic controller – part 1", IEEE Trans. Systems, Man and Cybernetics, Vol. 20, No. 2, pp. 404-418.

Lee, C. C., (1990b), "Fuzzy logic in control systems: fuzzy logic controller – part 2", IEEE Trans. Systems, Man and Cybernetics, Vol. 20, No. 2, pp. 419-435.

Leonhard, N. E., and Levine, W. S., (1995), "Using MATLAB to analyze and design control systems", 2nd Edition, Addison-Wesley Publishing Company.

Leonhard, W., (1986), "Microcomputer control of high dynamic performance ac-drives-a survey", Automatica, Vol. 22, No. 1, pp. 1-19.

Leonhard, W., (1988), "Adjustable-speed ac drives", Proc. of IEEE, Vol. 76, No. 4, pp. 455-471.

Liaw, C. M., Kung, Y. S., and Wu, C. M., (1991), "Design and implementation of a high performance field-oriented induction motor drive", IEEE Trans. Indus. Elect., Vol. 38, No. 4, pp. 275-282.

Liaw, C., and Wang, J., (1991), "Design and implementation of fuzzy controller for a high performance induction motor drive", IEEE Trans. Systems, Man and Cybernetics, Vol. 21, No. 4, pp. 921-929.

Lim, K. W., Low, T. S., Rahman, M. F., and Wee, L. B., (1991), "A discrete time variable structure controller for brushless dc motor drive", IEEE Trans. Indus. Elect., Vol. 38, No. 2, pp. 102-107.

Lin, F. J., Fung, R. F., and Wang, Y. C., (1997), "Sliding mode and fuzzy control of toggle mechanism using PM synchronous servomotor drive", IEE Proc. Control Theory App., Vol. 144, No. 5, pp. 393-402.

Lin, F.-J., and Chiu, S.-L., (1998), "Adaptive fuzzy sliding-mode control for PM synchronous servomotor drives", IEE Proc. Control Theory App., Vol. 145, No. 1, pp. 63-72.

Lin, S. C., and Tsai, S. C., (1984), "A microprocessor-based incremental servo system with variable structure", IEEE Trans. Indus. Elect., Vol. IE-31, No. 4, pp. 313-316.

Lipo, T. A., (1979), "Simulation of a current source inverter drive", IEEE Trans. Ind. Electron. and Control Inst., Vol. IECI-26, No. 2, pp. 98-103.

Lipo, T. A., and Cornell, E. P., (1975), "State-variable steady-state analysis of a controlled current induction motor drive", IEEE Trans. Ind. Appl., Vol. IA-11, No. 6, pp. 704-712.

Lipo, T. A., and Cosoli, A., (1984), "Modeling and simulation of induction motors with saturable leakage reactances", IEEE Trans. Ind. Appl., Vol. IA-20, No. 1, pp. 180-189.

Lipo, T. A., and Krause, P. C., (1969), "Stability analysis of a rectifier-inverter induction motor drive", IEEE Trans. Power App. and Systems, Vol. PAS-88, No. 5, pp. 55-66.

Lipo, T. A., Krause, P. C., and Jordan, H. E., (1969), "Harmonic torque and speed pulsation in a rectifier-inverter induction motor drive", IEEE Trans. Power App. and Systems, Vol. PAS-88, No. 5, pp. 579-587.

Liu, T. H., and Lin, M. T., (1996), "A fuzzy sliding-mode controller design for a synchronous reluctance motor drive", IEEE Trans. Aerospace and Electronic Systems, Vol. 32, No.3, pp. 1065-1075.

Lo, J. C., and Kuo, Y. H., (1998), "De-coupled fuzzy sliding-mode control", IEEE Trans. Fuzzy Systems, Vol. 6, No. 3, pp. 426-435.

Lorenz, R. D., (1986), "Tuning of field-oriented induction motor controllers for high-performance applications", IEEE Trans. Indus. App., Vol. IA-22, No. 2, pp. 293-297.

Mamdani, E. H., (1974), "Application of fuzzy algorithms for control of simple dynamic plant", Proc. IEE, Control and Science, Vol. 121, No. 12, pp. 1585-1588.

Mendel, J. M, (1995), "Fuzzy logic systems for engineering: a tutorial", Proceedings of the IEEE, Engineering App. of Fuzzy Logic, pp. 345-377.

Miki, I., Nakao, O., and Nishiyama, S., (1991), "A new simplified current control method for field-oriented induction motor drives", IEEE Trans. Indus. App., Vol. 27, No. 6, pp. 1081-1086.

Mir, S. A., Elbuluk, M. E., and Zinger, D. S., (1994), "Fuzzy implementation of direct self control of induction machines", IEEE Trans. Industry Applications, Vol. 30, No. 3, pp. 729-735.

Mittal, M., and Ahmed, N. U., (1983), "Time domain modeling and digital simulation of variable-frequency ac motor speed control using PLL technique", IEEE Trans. Industry Applications, Vol. IA-19, No. 2, pp. 174-180.

Moffat, R., and Sen, P. C., Younker, R., and Bayoumi, M. M., (1979), "Digital phase-locked loop for induction motor speed control", IEEE Trans. Industry Applications, Vol. IA-15, No. 2, pp. 176-182.

Mohan, N., Undeland, T. M., and Robbins, W. P., (1995), "Power electronics converters, application, and design", John Wiley and Sons Inc., 2nd edition.

Mokrytzki, B., (1968), "The controlled slip static inverter drive", IEEE Trans. Indus. and General App., Vol. IGA-4, No. 3, pp. 312-317.

Murphy, J. M. D., and Turnbull, F. G., (1988), "Power electronic control of AC motors", Pergamon Press.

Mutoh, N., Nagase, H., Sakai, K., and Ninomiya, H., (1986), "High-response digital speed-control system for induction motors", IEEE Trans. Indus. Elect., Vol. IE-33, No. 1, pp. 52-58.

Nabae, A., Otsuka, K., Uchino, H., and Kurosawa, R., (1980), "An approach to flux control of induction motors operated with variable-frequency power supply", IEEE Trans. Indus. App., Vol. IA-16, No. 3, pp. 342-349.

Nishimoto, M., Dixon, J. W., Kulkarni, A. B., and Ooi, B. T., (1987), "An integral controlled-current PWM rectifier chopper link for sliding mode position control", IEEE Trans. Indus. App., Vol. IA-23, No. 5, pp. 894-900.

Nonaka, S., and Neba, Y., (1994), "Current regulated PWM-CSI induction motor drive system without a speed sensor", IEEE Trans. Indus. App., Vol. 30, No. 1, pp. 116-124.

Nordin, K. B., Novotny, D. W., and Zinger, D. S, (1985), "The influence of motor parameter deviation in feedforward field orientation drive systems", IEEE Trans. Indus. App., Vol. IA-21, No. 4, pp. 1009-1015.

Ogata, K., (1988), "Modern control engineering", Prentice Hall. Inc., Englewood Cliffs, N. J.

Ohmae, T., Matsuda, T., Kamiyama, K., and Tachikawa, M., (1982), "A microprocessor-controlled high-accuracy wide-range speed regulator for motor drives", IEEE Trans. Industrial Electronics, Vol. IE-29, No. 3, pp. 207-212.

Pappis, C. P., and Mamdani, E. H., (1977), "A fuzzy logic controller for a traffic junction", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-7, No. 10, pp. 707-717.

Plunkett, A. B., (1977), "Direct flux and torque regulation in a PWM inverter-induction motor drive", IEEE Trans. Indus. App., Vol. IA-13, No. 2, pp. 139-146.

Rajan, S. D., Jacovides, L. J., and Lewis, W. A., (1974), "Digital simulation of a high-performance AC drive system –part 1", IEEE Trans. Ind. Appl., Vol. IA-10, No. 3, pp. 391-396.

Rajan, S. D., Jacovides, L. J., and Lewis, W. A., (1974), "Digital simulation of a high-performance AC drive system –part 2", IEEE Trans. Ind. Appl., Vol. IA-10, No. 3, pp. 397-402.

Rajashekara K. S., and Vithayathil J., (1982), "Microprocessor based sinusoidal PWM inverter by DMA transfer", IEEE Trans. Ind. Electron., Vol. IE-29, No. 1, pp. 46-51.

Rashid, M. H., (1993), "Power electronics, circuits, devices and applications", Prentice-hall international, Inc., 2nd Edition.

Robertson, S. T., and Hebbar, K. M., (1969), "Digital model for three-phase induction machines", IEEE Trans. Power App. and Systems, Vol. PAS-88, No. 11, pp. 1624-1634.

Rosseto, L., Spiazzi, G., Tenti, P., Fabiano, B., and Licitra, C., (1994), "Fast-response high-quality rectifier with sliding mode control", IEEE Trans. Power Elect., Vol. 9, No. 2, pp. 146-151.

Sabanovic, A., and Bilalovic, F., (1989), "Sliding mode control of ac drives", IEEE Trans. Indus. App., Vol. 25, No. 1, pp. 70-75.

Sabanovic, A., and Izosimov, D. B., (1981), "Application of sliding modes to induction motor control", IEEE Trans. Indus. App., Vol. IA-17, No. 1, pp. 41-49.

Saito, K., Kamiyama, K., Ohmae, T., and Matsuda, T., (1988), "A microprocessor-controlled speed regulator with instantaneous speed estimation for motor drives", IEEE Trans. Industrial Electronics, Vol. 35, No. 1, pp. 95-99.

Saito, K., Nagasaki, H., Doi, K., Matsuka, S, and Kamiyama, K., (1984), "Application of fully digital speed regulators to tandem cold mills", IEEE Trans. Industrial Applications, Vol. IA-20, No. 4, pp. 785-794.

Salihi, J. T., (1969), "Simulation of controlled-slip variable-speed induction motor drive systems", IEEE Trans. Ind. and General Appl., Vol. IGA-5, No. 2, pp. 149-157.

Sarma, M. S., (1986), "Electric machines steady-state theory and dynamic performance", West Publishing Co.

Sarpturk, S. Z., Istefanopulos, Y., and Kaynak, O., (1987), "On the stability of discrete-time sliding control systems", IEEE Trans. Autom. Cont., Vol. AC-32, No. 10, pp. 930-932.

Schauder, C. D., and Caddy, Y., (1994), "Current regulated PWM-CSI induction motor drive system without a speed sensor", IEEE Trans. Indus. App., Vol. 30, No. 1, pp. 116-124.

Schauder, C. D., Choo, F. H., and Robert, M. T., (1983), "High performance torque-controlled induction motor drives", IEEE Trans. Indus. App., Vol. IA-19, No. 3, pp. 349-355.

Schwartz, D. G., Klir, G. J., Lewis, H. W., and Ezawa, Y., (1994), "Applications of fuzzy sets and approximate reasoning", IEEE Proc. Consumer Electronics, Vol. 82, No. 4, pp. 482-498.

Song, Y.-H., and Johns, A. T., (1998), "Application of fuzzy logic in power systems", IEE Power Engineering Journal, Vol. 12, No. 4, pp. 185-191.

Sousa, G. C., Bose, B. K., and Cleland, J. G., (1995), "Fuzzy logic based on-line efficiency optimization control of an indirect vector-controlled induction motor drive", IEEE Trans. Industrial Electronics, Vol. 42, No. 2, pp. 192-198.

Steven, R. E., "Electrical machines and power electronics", (1983), Van Nostrand Reinhold (UK) Co. Ltd.

Su, C. Y., Leung, T. P., and Stepanenko, Y., (1993), "Real-time implementation of regressor-based sliding mode control algorithm for robotic manipulators", IEEE Trans. Indus. Elect., Vol. 40, No. 1, pp. 71-79.

Suyitno, A., Fujikawa, J., Kobayashi, H., and Dote, Y., (1993), "Variable-structured robust controller by fuzzy logic for servomotors", IEEE Trans. Indus. Elect., Vol. 40, No. 1, pp. 80-87.

Takagi, T., and Sugeno, M., (1985), "Fuzzy identification of systems and its applications to modelling and control", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-15, No. 1, pp. 116-132.

Takahashi, I., and Ohmori, Y., (1989), "High-performance direct torque control of an induction motor", IEEE Trans. Indus. App., Vol. 25, No. 2, pp. 257-264.

Tez, E., S., (1995), "A simple understanding of field-orientation for AC motor control", IEE Colloq. Vector control and direct torque control of induction motors, pp. 3/1-3/4.

Thorborg, K., (1988), "Power electronics", Prentice Hall.

Tzafestas, S., and Papanikolopoulos, N. P., (1990), "Incremental fuzzy expert PID control", IEEE Trans. Industrial Electronics, Vol. 37, No. 5, pp. 365-371.

Utkin, V. I., (1977), "Variable structure systems with sliding modes", IEEE Trans. Autom. Cont., Vol. AC-22, No. 2, pp. 212-222.

Utkin, V. I., (1993), "Sliding mode control design principles and applications to electric drives", IEEE Trans. Indus. Elect., Vol. 40, No. 1, pp. 23-35.

Vas, P., (1990), "Vector control of AC machines", Clarendon Press, Oxford.

Vas, P., and Stronach, A. F., (1996), "Adaptive fuzzy-neural DSP control of high performance drives", Proc. IEE International Power Electronics and Variable Speed Drives Conference (PEVD), London, UK, Conference Publications No. 429, pp. 424-429.

Vas, P., Stronach, A., and Neuroth, M., (1997), "Full fuzzy control of a DSP-based high performance induction motor drive", IEEE Proc. Control Theory App., Vol. 144, No. 5, pp. 361-368.

Wade, S., Dunnigan, M. W., and Williams, B. W., (1995), "Comparison of stochastic and deterministic parameter identification algorithms for indirect vector control", IEE Colloq. Vector control and direct torque control of induction motors, pp. 2/1-2/5.

Wang, J. B., and Liaw, C. M., (1997), "Indirect field-oriented induction motor drive with fuzzy detuning correction and efficiency optimisation controls", IEE Proc. Electr. Power Applications, Vol. 144, No. 1, pp. 37-45.

Wang, L., and Mendel, J. M., (1992), "Generating fuzzy rules by learning from examples", IEEE Trans. Systems, Man and Cybernetics, Vol. 22, No. 6, pp. 1414-1427.

Wu, J. C., and Liu, T. S., (1996), "A sliding-mode approach to fuzzy control design", IEEE Trans. Control Systems Technology, Vol. 4, No. 2, pp. 141-151.

Wu, Z. K., and Strangas, E. G., (1988), "Feed forward field orientation control of an induction motor using a PWM voltage source inverter and standardised single-board computers", IEEE Trans. Indus. Elect., Vol. 35, No. 1, pp. 75-79.

Xu, J. X., Hashimoto, H., Slotine, J. E., and Arai, Y., and Harashima, F., (1989), "Implementation of VSS control to robotic manipulator-smoothing modification", IEEE Trans. Indus. Elect., Vol. 36, No. 3, pp. 321-329.

Yi, S. Y., and Chung, M. J., (1998), "Robustness of fuzzy logic control for an uncertain dynamic system", IEEE Trans. Fuzzy Systems, Vol. 6, No. 2, pp. 216-225.

Yuvarajan, S., Ramaswami B., and Subrahmanyam, V., (1980), "Analysis of a current-controlled inverter- fed induction motor drive using digital simulation", IEEE Trans. Ind. Electron. and Control Inst., Vol. IECI-27, No. 2, pp. 67-76.

Zadeh, L. A., (1965), "Fuzzy sets", Information and Control, Vol. 8, pp. 338-353.

Zadeh, L. A., (1973a), "Syllogistic reasoning in fuzzy logic and its application to the usuality and reasoning with dispositions", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-15, No. 6, 1985, pp. 754-763.

Zadeh, L. A., (1973b), "Outline of a new approach to the analysis of complex systems and decision processes", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-3, No. 1, pp. 28-44.

Zadeh, L. A., Fu, K. S., Tanaka, K., and Shimura, M., (1975), "Fuzzy sets, and their applications to cognitive and decision processes", Academic Press Inc.

Zhen, L., and Xu, L., (1998), "On-line fuzzy tuning of indirect field-oriented induction machine drives", IEEE Trans. Power Electronics, Vol. 13, No. 1, pp. 134-141.

Zuckerberger A., Alexandrovitz A., (1986), "Determination of commutation sequence with a view to eliminating harmonics in microprocessor-controlled PWM voltage inverter", IEEE Trans. Ind. Electron., Vol. IE-33, No. 3, pp. 262-270.

Zuckerberger, A., Weinstock, D., and Alexandrovitz, A., (1996), "Simulation of three-phase loaded matrix converter", IEE Proc. Power Appl., Vol. 143, No. 4, pp. 294-300.

"16-bit embedded controllers", (1990), Intel Data Book.

"SIMULINK User's Guide", (1992), The Math Works Inc.

"The student edition of SIMULINK, User's Guide", (1996), The Math Works Inc., Prentice Hall, Englewood Cliffs, NJ.

APPENDIX-A

Determination of induction motor parameters

DC test:

V_{dc} = 18.8V

I_{dc} = 5A

No-load test:

V_{line} = 410V, $V_{NL} = \frac{410}{\sqrt{3}} = 236.7\text{V}$

I_{line} = 4.3A, I_{NL} = 4.3A

P_{3phase} = 1030W, $P_{NL} = \frac{1030}{3} = 343\text{W}$

Locked rotor test:

V_{line} = 79V, $V_{LR} = \frac{79}{\sqrt{3}} = 45.6\text{V}$

I_{line} = 5A, I_{LR} = 5A

P_{3phase} = 345.5W, $P_{LR} = \frac{345.5}{3} = 115\text{W}$

- Determination of R_s:

$$R_{dc} = \frac{V_{dc}}{I_{dc}} = \frac{18.8V}{5A} = 3.76\Omega$$

$$R_s = \frac{R_{dc}}{2} = 1.88\Omega \text{ / phase}$$

- Determination of R_r:

$$Z_{LR} = \frac{V_{LR}}{I_{LR}} = \frac{45.6V}{5A} = 9.1\Omega$$

$$R_{LR} = \frac{P_{LR}}{(I_{LR})^2} = \frac{115W}{(5A)^2} = 4.6\Omega$$

$$R_r = R_{LR} - R_s = 4.6 - 1.88 = 2.72\Omega \text{ / phase}$$

- Determination of L_{ls} and L_{lr}:

$$X_{LR} = \sqrt{Z_{LR}^2 - R_{LR}^2} = \sqrt{(9.1)^2 - (4.6)^2} = 7.85\Omega$$

$$X_{ls} = 0.5 * X_{LR} = 3.926\Omega \text{ / phase}$$

$$X_{lr} = 0.5 * X_{LR} = 3.926\Omega \text{ / phase}$$

$$L_{ls} = \frac{X_{ls}}{2\pi f} = \frac{3.926}{2\pi * 50} = 0.0125\text{H / phase}$$

$$L_{lr} = \frac{X_{lr}}{2\pi f} = \frac{3.926}{2\pi * 50} = 0.0125\text{H / phase}$$

- Determination of L_m:

$$S_{NL} = V_{NL} * I_{NL} = 236.7\text{V} * 4.3\text{A} = 1018\text{VA}$$

$$Q_{NL} = \sqrt{S_{NL}^2 - P_{NL}^2} = \sqrt{(1018)^2 - (343)^2} = 958.3\text{VAr}$$

$$X_{NL} = \frac{Q_{NL}}{(I_{NL})^2} = \frac{958.3}{(4.3)^2} = 51.828\Omega$$

$$X_m = X_{NL} - X_s = 51.828 - 3.926 = 47.902\Omega$$

$$L_m = \frac{X_m}{2\pi f} = \frac{47.902}{2\pi * 50} = 0.1524\text{H}$$

APPENDIX-B

An overview of the 80C196KC microcontroller system

The 80C196KC is a 16-bit, 16-MHz CHMOS microcontroller designed especially to handle high speed calculations required for applications such as motors, electronic engines, suspension, anti-lock braking control and hard disk drives. It shares a common architecture and instruction set with other members of the MCS-96 family. The 80C196KC has a 64K byte address space, where 256 bytes are used for Special Function Register (SFR) and Register File. The main components of this microcontroller include a CPU, several types of memory, five 8-bit I/O ports, and several internal peripheral devices. The peripherals include a serial I/O port, an A/D converter, a High-Speed Input/Output (HSIO) unit and two timers. Figure B-1 shows a block diagram of the 80C196KC microcontroller device.

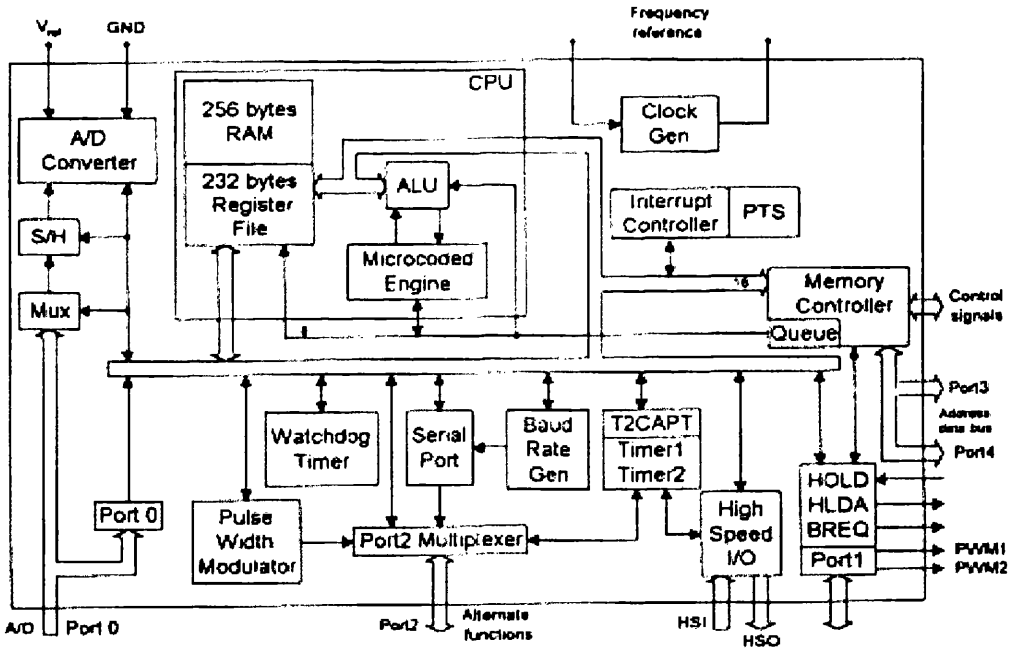


Figure B-1: Block diagram of the 80C196KC microcontroller device.

The 80C196KC architecture supports a variety of signed or unsigned data (operand) types such as Byte (8-bits), Word (16-bits) and Double-Word (32-bits), which are likely to be useful in control applications. These operands are usually accessed within the 64K address space with six basic addressing modes, which are Register-Direct Addressing, Indirect Addressing, Indirect with Auto-

Increment Addressing, Immediate Addressing, Short-Indexed Addressing and Long-Indexed addressing. Two other useful modes are Zero Register Addressing and Stack Pointer Register Addressing.

B.1 Central Processing Unit (CPU) and Memory:

The Central Processing Unit (CPU) consists of a Register Arithmetic-Logic Unit (RALU) and a 512-byte Register File (the lower 24 bytes of the Register File are reserved for SFRs and cannot be used as general purpose RAM). The RALU does not use an accumulator, but instead it operates directly on the lower 256 bytes of the Register File, which essentially provides 256 "accumulators". Other areas of memory are accessed via the memory controller, which includes a 4-byte pre-fetch queue to speed program execution, and a bus controller. The bus controller manages access to four areas of memory which are the 256-byte Internal Ram (for code or data), 16K bytes of on-chip ROM/EPROM, Special Function Registers (SFRs) for the on-chip peripherals, and external memory.

B.2 Interrupts:

A microcontroller's primary function is to provide a real-time control of an instrument or device. The interrupt control circuitry within a microcontroller permits real-time events to control program flow. When an event generates an interrupt, the CPU services the interrupt routine before executing the next instruction. An internal peripheral, an external signal, or an instruction can request an interrupt. In the simplest case, the 80C196KC receives the request, performs the service, and returns to the task that was interrupted. Some of these interrupts include a Software Timer Interrupt, a HSI.0 Interrupt, a HSO Interrupt and an A/D Conversion Complete Interrupt.

B.3 High-Speed Input/Output Unit:

The 80C196KC contains four peripherals that together form a flexible, timer/counter-based, high-speed input/output (HSIO) unit. These are Timer 1, Timer 2, the High-Speed Input (HSI) module, and the High-Speed Output (HSO) module. The HSIO can measure pulse widths, generate waveforms, and create periodic interrupts with very little CPU overhead.

B.3.1 Timer 1 Functional Overview:

Timer 1 is a standard 16-bit, free-running timer that is increased once every eight state times (8 μsec). Timer 1 is always the time base for the HSI module, but it can also be selected as the time base for the HSO module. The two bytes of the Timer 1 Register contain the Timer 1 count, and it can cause an interrupt when it overflows.

B.3.2 Timer 2 Functional Overview:

Timer 2 is a programmable 16-bit counter that can be used as the time base for the HSO module, as an up/down counter, or as an extra counter. It can also capture external events, and it can be clocked either internally or externally. The maximum count rate for Timer 2 is once every state time (fast increment mode) or every eight state times (normal mode). The two bytes of the Timer 2 Register contain the value of Timer 2.

B.3.3 High-Speed Input Module:

The High-Speed Input (HSI) module monitors four external input pins (HSI.0-HSI.3). When a predefined event occurs on one or more of the pins, the HSI module records the current Timer 1 count value along with a status bit for each input. The HSI can be programmed to capture each positive transition, each negative transition, every transition (both positive and negative), or every eight positive transitions. This variety enables the HSI module to measure many types of inputs such as pulse widths, periods and phase differences. Recording every eighth positive transition allows faster pulse rates to be measured and counted.

B.3.4 High-Speed Output Module:

The High-Speed Output (HSO) module triggers events at specified times using either Timer 1 or Timer 2 as the time reference. These programmable events include starting an A/D conversion, resetting Timer 2, generating up to four software time delays, and setting or clearing one or more of the HSO outputs (HSO.0-HSO.5). The HSO module stores up to eight pending events and their specified times in a Content-Addressable Memory (CAM) file. Figure B-2, shows the block diagram of the HSO module.

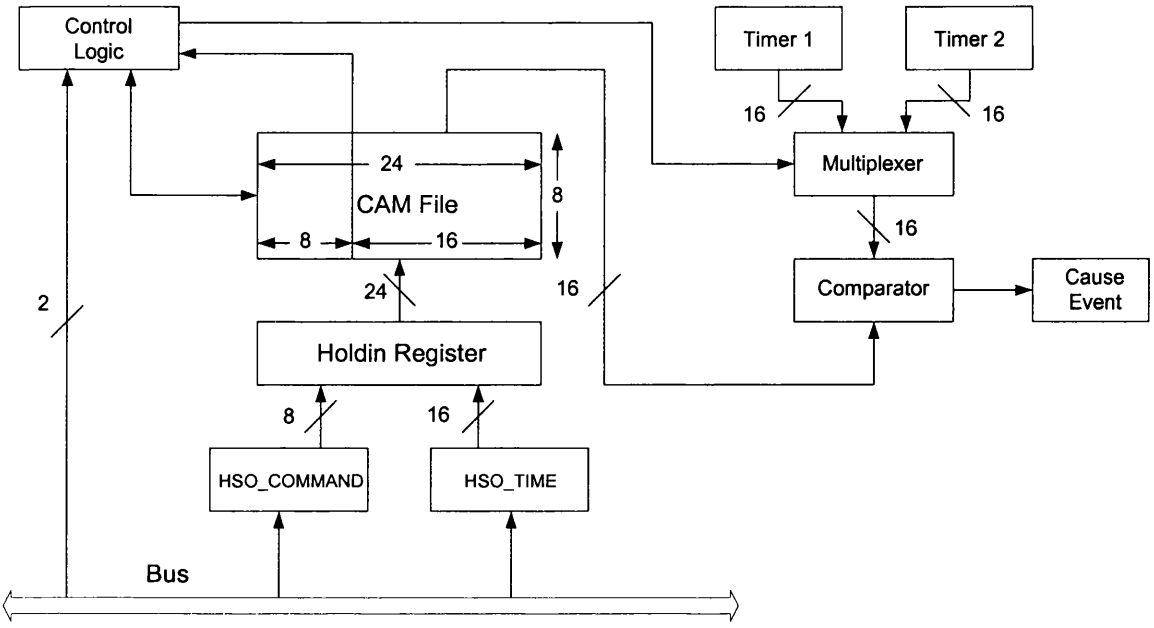


Figure B-2: HSO block diagram.

The CAM file is the main component of the HSO module. This file stores up to eight commands. Each CAM entry is 24 bits wide. Sixteen bits are loaded from the HSO_TIME Register to specify the action time, and eight bits are loaded from the HSO_COMMAND Register to specify the nature of the action, whether it will generate an interrupt, and whether Timer 1 or Timer 2 is the reference. The HSO compares all eight CAM entries with the timer value before triggering an event. It takes one state time (0.125 μsec) to compare each CAM entry, so eight state times are needed for a complete CAM search. The HSO triggers the event when the selected timer matches the programmed time value.

B.4 A/D Functional Overview:

The A/D converter module converts an analogue input to the microcontroller into either 10-bit or faster 8-bit digital representation. The main parts of the A/D converter module include eight analogue inputs (ACH0-ACH7), 8-channel multiplexer to select one of the eight input channels, and 10-bit successive approximation A/D converter. The A/D conversion is controlled by an AD-COMMAND Register, which selects the A/D channel to be converted, determines whether the conversion starts immediately or with an HSO command, and selects either the 8-bit or 10-bit conversion mode. The result of the conversion is then stored into an AD_RESULT Register, which

consists of two bytes. The high byte contains the eight most-significant bits from the conversion, while the lower byte indicates the A/D channel number that was used for the conversion, whether a conversion is currently in progress, and contains the two least-significant bits from a ten-bit conversion.

APPENDIX-C

M/T method for digital speed measurement and the operation of EX-OR logic gate

The basic principle of the M/T method is illustrated in Figure C-1, in which the detection period T_d is determined by synchronising the first encoder pulse after the pre-defined sampling period T_s. Thus, the measured variable of the motor speed (ω_r) in radians per seconds during T_d can be defined by the following formula,

$$\omega_r = \frac{\theta_r}{T_d} \quad (rad/sec) \tag{C.1}$$

where θ_r is the angular displacement of the motor shaft during T_d. Equation (C.1) can be rewritten to express the motor speed in rev/min as follows,

$$\omega_r = \frac{60 * \theta_r}{2\pi * T_d} \quad (\text{rev/min}) \tag{C.2}$$

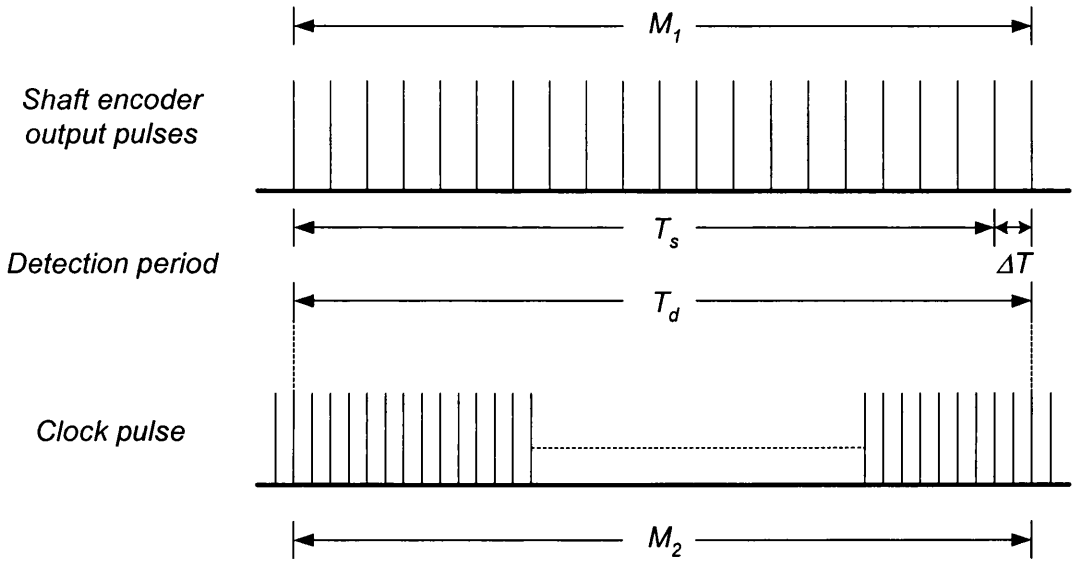


Figure C-1: Principle of the M/T digital speed measurement.

If M_1 output pulses generated from the shaft encoder are counted within the detection period T_d, then the angular displacement θ_r can be expressed as,

$$\theta_r = \frac{2\pi * M_1}{P_S} \tag{C.3}$$

where P_S is number of pulses that the shaft encoder is capable of generating per one rotor revolution. On the other side, for the microcontroller implementation, the detection time of T_d is digitised with the clock frequency f_s (hertz) of *Timer_1*, which is clocked at every 1 μs. Thus, T_d can be digitally expressed as,

$$T_d = \frac{M_2}{f_s} \tag{C.4}$$

where M_2 represents the digital value of the speed detection period, which is determined by subtracting the recorded timing value of *Timer_1* at the start of T_d from that recorded at the end. if Equations (C.3) and (C.4) are substituted in (C.2), it yields the general form of the M/T method for digital speed measurement as follows,

$$\omega_r = \frac{60\, f_s\, M_1}{P_s\, M_2} \text{ (rev/min)} \tag{C.5}$$

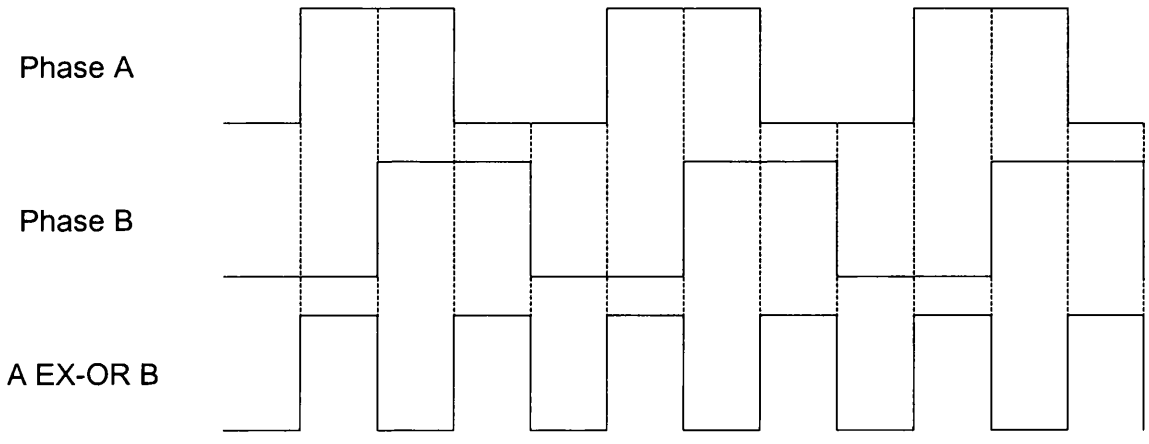


Figure C-2: Timing diagram for the EX-OR logic circuit.

On the other hand, an improved accuracy of the speed measurement is made possible by doubling the number of pulses that generated from the shaft encoder per one revolution. This is achieved by feeding the two output signals (phases A and B) from the encoder to an external logic circuit utilising an Exclusive-OR (EX-OR) logic gate. Consequently, the resulted output would be a pulse train signal with twice the frequency of phase A or phase B, as shown in the timing diagram of Figure C-2. The fabrication of the external logic circuit is shown in Figure C-3, as proposed by Guo (1994).

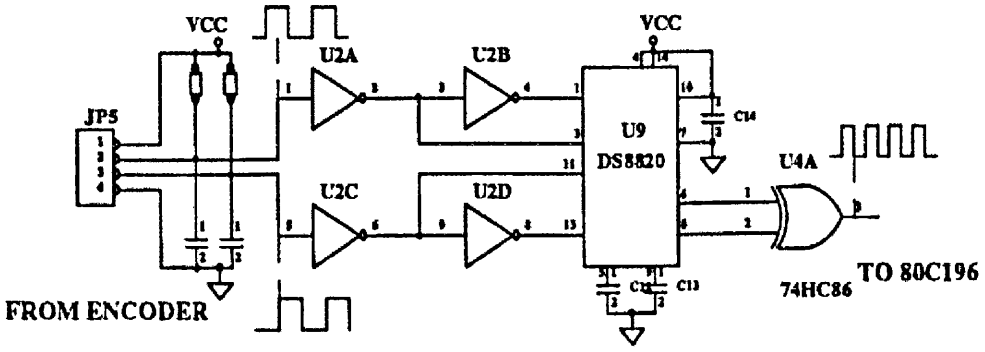


Figure C-3: External logic circuit for the shaft encoder.

APPENDIX-D

An overview of the Moving Average Filter Design

Performing some common processing operation on a sequence of data may be considered as filtering the raw data. For example, taking the mean of a finite sequence $x_1, x_2, \ldots x_N$,

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{D.1}$$

is an effect a very simple low-pass filtering operation that passes a single frequency component, the zero-frequency, or the mean. Indeed, all the data must be processed before the single number $\bar{x}$ is obtained. It is useful for removing low-frequency trends from the data. However, for long data processing, the mean as defined by (D.1) may have a minute practical significance. Instead, what is required is a running estimate of the mean, or so-called Moving Average. Taking the moving average implies averaging the sequence of length N in successive sub-sequences of length $m < N$. Applying this method with $m = 4$ is presented in Figure E-1, in which a "*window*" of length m is moved along the data.

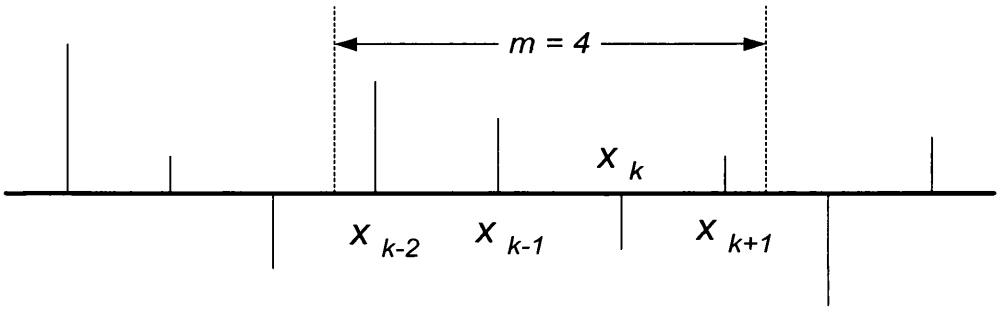


Figure D-1: The moving average operation.

The moving average is thus defined by the following expression,

$$y_k = \overline{x_k} = \frac{1}{m}\sum_{i=0}^{m-1} x_{k-i}\,, \qquad m-1 \le k \le N \tag{D.2}$$

The frequency characteristics of this operation can be determined by taking the z-transform of Equation (D.2). Thus, the transfer function is,

$$H(z) = \frac{1}{m} \sum_{i=0}^{m-1} z^{-i} \tag{D.3}$$

Setting "z" equal to $e^{j\omega T}$ in Equation (D.3) gives the frequency response as,

$$H(j\omega T) = \frac{1}{m} \frac{\sin \frac{m\omega T}{2}}{\sin \frac{\omega T}{2}} \tag{D.4}$$

A plot of this function is shown in Figure D-2. The min-lobe width measured from $-\omega_s/m$ to ω_s/m, is $2\omega_s/m$. Examination of this figure, reveals that the moving average operation can be regarded as a low pass filtering with a cut-off frequency $\omega_c = \omega_s/(2m)$, at which point the magnitude response is down about 3 to 4 dB, depending on the value of m.

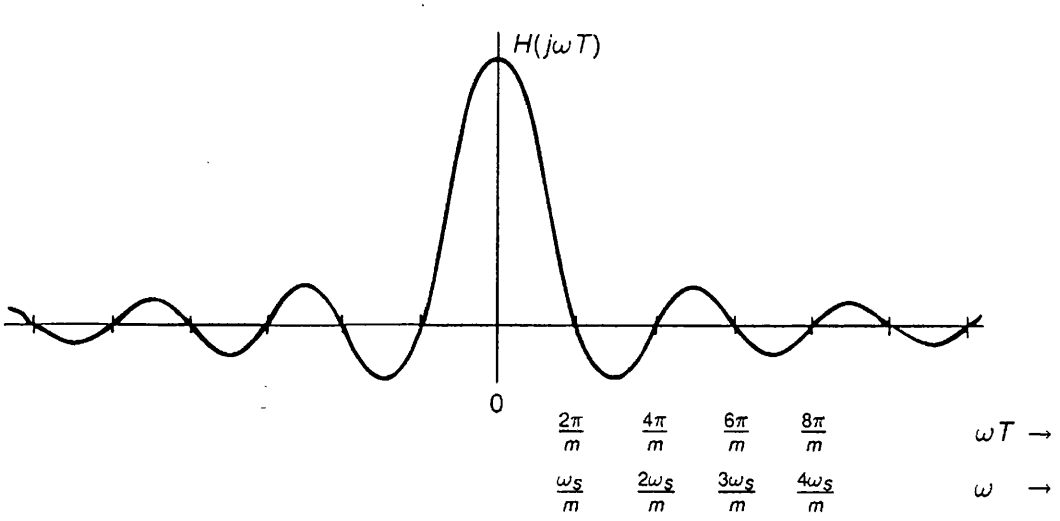


Figure D-2: The frequency response for moving-average processing.

APPENDIX-E

List of Source Code For Practical Implementation of Fuzzy Logic Speed Control of Induction Motor

```
$TITLE('Fuzzy-cont.ASM') ; Fuzzy Logic Controller
$INCLUDE(8096.INC)

RSEG at 40H

        time0:                  dsw 1
        table1:                 dsw 1
        table2:                 dsw 1

        ton:                        dsl 1
        ton_lo  equ     ton         :word
        ton_hi  equ     (ton+2)     :word

        toff:                       dsl 1
        toff_lo  equ     toff       :word
        toff_hi  equ     (toff+2)   :word

        Fr:                     dsw 1
        conv_const:             dsw 1
        const_per:              dsw 1
        time_ref1:              dsw 1
        time_ref2:              dsw 1
        time_ref3:              dsw 1

        Mr:                     dsw 1
        Tc:                     dsw 1
        Tc2_deg:                dsb 1
        k:                      dsw 1

        var2:                   dsl 1
        var2_lo  equ     var2    :word
        var2_hi  equ     (var2+2):word

        ax:                     dsw 1
        al       equ       ax    :byte
        ah       equ       (ax+1):byte

        pulse_on:          dsw 1
        pulse_off:         dsw 1
        counter1:          dsw 1
        soft_flg:          dsb 1
        start_soft:        dsb 1
        dely:              dsb 1
        start_flg:         dsb 1
        swap_flg:          dsb 1
        test1_flg:         dsb 1
        table3:            dsw 1
        table4:            dsw 1
        table5:            dsw 1
```

```
speed_table:        dsw 1
accelr_table:       dsw 1
per1_flg:           dsb 1
Fm:                 dsw 1

Tc_sec:             dsw 1
const:              dsw 1

Tc_flg:             dsb 1
period:             dsw 1
sw_exp_flg:         dsb 1
no_pulses:          dsw 1
Ts:                 dsw 1
per_st:             dsw 1
per_end:            dsw 1
sw_set_flg:         dsb 1
first_flg:          dsb 1
hsi_flg:            dsb 1
number:             dsw 1
num2:               dsw 1
set_speed:          dsw 1
deror:              dsw 1
old_error:          dsw 1
mod_rat:            dsw 1
speed1:             dsw 1
speed2:             dsw 1
speed3:             dsw 1
speed4:             dsw 1
speed_last:         dsw 1
speed_last4:        dsw 1
speed_last3:        dsw 1
speed_last2:        dsw 1
speed_last1:        dsw 1
speed_final:        dsw 1
speed_old:          dsw 1
P:                  dsw 1
angle_sec1:         dsw 1
angle_deg1:         dsw 1
angle_deg2:         dsw 1
pointer1:           dsw 1
pointer2:           dsw 1
ack_change:         dsb 1
ack1_change:        dsb 1
P_third:            dsw 1
P_60:               dsw 1
P_120:              dsw 1
U_int:              dsw 1
U_int_old:          dsw 1
U:                  dsw 1
U_F:                dsw 1
U_F_old:            dsw 1
result:             dsb 1
de_pointer:         dsw 1
e_pointer:          dsw 1
index:              dsw 1
wslip:              dsw 1
error:              dsw 1
```

```
        Rot_freq:           dsw 1
        stop_close:         dsb 1
        compl_flg:          dsb 1
        var:                    dsl 1
        var_lo  equ     var     :word
        var_hi  equu    (var+2):word

        const1:             dsw 1
        table_flg:          dsb 1
        start_tab:          dsb 1
        wslip_table:        dsw 1


CSEG at 2000H

        dcw     not_used
        dcw     ad_int
        dcw     not_used
        dcw     hso_int
        dcw     hsi_int
        dcw     swt_int


;******************-------------------------------------*****************;
;******************|            Closed-Loop             |*****************;
;******************|  (V/f) control with Fuzzy logic    |*****************;
;******************|    including sinusoidal PWM        |*****************;
;******************|    generation with M/T speed       |*****************;
;******************|     detection and MA-Filter        |*****************;
;******************-------------------------------------*****************;


CSEG at 2080H

        di
        ld      sp, #100H

;****************[ Selection of Fr at Fm = 50 Hz]*********************;
;*****************************************************************;
;*****  Fr      Fc      ***********************************************;
;*****  18      0.9 kHz ***********************************************;
;*****  30      1.5 kHz ***********************************************;
;*****  60      3.0 kHz ***********************************************;
;*****  90      4.5 kHz ***********************************************;
;*****  120     6.0 kHz ***********************************************;
;*****  180     9.0 kHz ***********************************************;
;*****************************************************************;

;************************[ INITIALISATIONS ]************************;
;*****************************************************************;

        ld      set_speed,#800
        ld      speed_old,#00

        ld      table1,#3000h   ;start table1 address in memory at 100h
        ld      table2,#3400h
                                ;F0=50Hz
```

```
                              ;Fr=Fc/F0 = no. 0f pulses per ac cycle
                              ;Fc=5Khz
                              ;Fr = 4000/50=100 (pulses per ac cycle)
        ld      Tc,#0         ;Tc=(1/Fc) Carrier period
        ld      Tc2_deg,#0
        ld      const_per,#360
        ld      const,#32768
        ld      const1,#24576
        ld      U_F_old,#0
        ld      time0,#0
        ld      time_ref1,#0  ;pulse width
        ld      k,#0          ;updated no. of pulses per half ac cycle
        ld      Fm,#62
        ld      counter1,#0
        ld      Fr,#18        ;no. of pulses/ac cycle(freq. ratio)
        ldb     soft_flg,#00b
        ldb     compl_flg,#00b
        ldb     start_soft,#01b
        ldb     dely,#0
        ldb     stop_close,#00b
        ldb     start_flg,#01b
        ldb     swap_flg,#01b
        ldb     test1_flg,#01b
        ld      ax,#0
        ldb     per1_flg,#00b
        ldb     Tc_flg,#00b
        ld      no_pulses,#0
        ldb     sw_exp_flg,#00b
        ldb     sw_set_flg,#00b
        ldb     first_flg,#00b
        ld      Ts,#5000
        ldb     hsi_flg,#00b
        ld      num2,#0
        ld      error,#0
        ld      speed_table,#6000h
        ld      accelr_table,#4500h
        ld      wslip_table,#4000h
        ldb     ack1_change,#00b
        ldb     ack_change,#00b
        ld      U_int_old,#0
        ldb     start_tab,#00b

        ldb     table_flg,#01b
        ldb     ioc1,#01010000b       ;enable hso.4 and hso.5 output
        ldb     ioc0,#00000001b       ;hsi.0 input enable
        ldb     hsi_mode,#00000010b
        ldb     ioc2,#10000000b

;*************************[ Cheack Set_speed]************************;
;********************************************************************;

        cmp     set_speed,#1000
        jge     limit_set
        br      init_table

LIMIT_SET:
        ld      set_speed,#1000
```

```
;************************[ Initialise Tables ]***********************;
;********************************************************************;

INIT_TABLE:

         st      0,[table1]+
         st      0,[table2]+
         cmp     table1,#3200h
         jle     init_table
         ld      table1,#3000h
         ld      table2,#3400h


;*********************[ Initialise Mod_ratio ]************************;
;********************************************************************;


        ldb     ad_command,#00001100b    ;start convertion now at ch4
        nop                              ;wait for conversion to start
        nop

CHECK:

        jbs     ad_result_lo,3,check     ;wait while A/D is busy;
        ldb     int_mask,#00111010b      ;enable hsi.0, swt, hso and A/D
        ei                               ;service interrupt now


CONVE_COMP:

        jbs     compl_flg,0,detect_speed
        ldb     ad_command,#00001100b    ;start convertion now at ch4
        br      conve_comp

;************************[ Start Calculations ]**********************;
;********************************************************************;


DETECT_SPEED:


        bbc     hsi_flg,0,update_result  ;calculate speed when swt
                                         ;expired
        ld      var_hi,#00
        mulu    var,number,#14648        ;(60*clock)/encoder resolution
                                         ;=30000
        divu    var,num2                 ;speed =(30000*pulses)/detect
                                         ;period


        ld      speed1,speed_last3

        ld      speed2,speed_last2
        ld      speed_last3,speed2

        ld      speed3,speed_last1
```

```
        ld      speed_last2,speed3


        ld      speed4,var_lo
        ld      speed_last1,speed4

        ld      speed_last4,speed_last1

        add     speed2,speed1
        add     speed3,speed2
        add     speed_last4,speed3
        ld      speed_final,speed_last4
        shr     speed_final,#2


QUIT:

        ldb     hsi_flg,#00b
        ld      speed_last,speed_final
        ld      old_error,error
        sub     error,set_speed,speed_last

        ld      var_hi,#00
        sub     var_lo,speed_old,speed_last
        shll    var,#5
        ld      deror,var_lo
        ld      speed_old,speed_last


SPEED_RESULT:

        jbc     start_tab,0,update_result
        st      speed_last,[speed_table]+
        st      wslip,[wslip_table]+
        st      deror,[accelr_table]+
        cmp     speed_table,#6400h
        jge     reset_table
        br      update_result

RESET_TABLE:

         ldb    start_tab,#00b

;************************[ Define P and pointers ]*******************;
;********************************************************************;

UPDATE_RESULT:

        bbs     ack_change,0,update_result
        bbs     ack1_change,0,soft_start
        add     P,Fr,#0                 ; No. of pulses per one cycle
        add     P_third,Fr,#0
        add     P_60,Fr,#0
        add     P_120,Fr,#0

        shr     P_60,#1
        divb    P_60,#3
```

```
        add     P_60,#2

        shr     P_120,#1
        divb    P_120,#3
        shl     P_120,#1
        sub     P_120,#2

        add     pointer1,Fr,#0
        shr     pointer1,#1
        divb    pointer1,#3
        shl     pointer1,#2

        add     pointer2,pointer1,#0
        shl     pointer2,#1

        shr     Fr,#1
        divb    P_third,#6              ; No. of pulses per 1/6 of a
cycle
        div     const_per,p
        ld      Tc,const_per            ; Tc(deg)   = 360(deg)/P
        shr     const_per,#1            ; Tc2(deg) = Tc(deg)/2
        ld      Tc2_deg,const_per

        ldb     ack1_change,#01b

;*****************************[ Soft-Start ]*************************;
;********************************************************************;

SOFT_START:

        bbs     soft_flg,0,closed_loop
        jbs     start_soft,0,dely_time
        br      loop


DELY_TIME:

        bbs     soft_flg,1,soft_mod
        incb    dely                    ;the soft_start loop cmprising
        cmpb    dely,#2                 ;a delay time for the motor to
        ble     start_calc1             ;start because of its inertia.


CONTUN_SOFT:

        ldb     dely,#0
        add     Fm,#6                    ;increment freq every loop by
                                         ;6 rad/sec
        cmp     Fm,#130                  ;check for freq limit.
        jge     soft_mod
        br      start_calc


SOFT_MOD:

        ld      Fm,#130                  ;limit frequency
        orb     soft_flg,#10b
```

```
        br      start_calc

;****************************[ Closed_Loop ]*************************;
;********************************************************************;

CLOSED_LOOP:

        bbs     start_flg,0,start_control  ;start calculation if flag
is set
        br      loop                       ;loop if no change in
mod_ratio

START_CONTROL:

;************************----------------------*********************;
;************************| Fuzzy Logic Contller |*********************;
;************************----------------------*********************;

;************************[ Define Column Index ]*********************;
;********************************************************************;

FUZZY_CONTROL:


        bbs     error,15,error_neg

;****************************[ Error_Positive ]**********************;
;********************************************************************;

ERROR_POS:

        cmp     error,#170              ;define 1st column in Fuzzy
matrix
        jge     index_170
        br      mid_pos

INDEX_170:

        ld      e_pointer,#24
        br      find_fuzz_row

MID_POS:
        cmp     error,#85
        jlt     low_pos

HIGH_POS:
        cmp     error,#126
        jlt     high_pos_2

HIGH_POS_1:
        cmp     error,#140
        jge     index_140
        ld      e_pointer,#20
        br      find_fuzz_row

INDEX_140:
        ld      e_pointer,#22
```

```
        br      find_fuzz_row

HIGH_POS_2:
        cmp     error,#113
        jge     index_113
        ld      e_pointer,#18
        br      find_fuzz_row

INDEX_113:
        ld      e_pointer,#20
        br      find_fuzz_row

LOW_POS:
        cmp     error,#36
        jlt     low_pos_2

LOW_POS_1:
        cmp     error,#57
        jge     index_57
        ld      e_pointer,#14
        br      find_fuzz_row

INDEX_57:
        ld      e_pointer,#16
        br      find_fuzz_row

LOW_POS_2:
        cmp     error,#5
        jge     index_5
        ld      e_pointer,#12
        br      find_fuzz_row

INDEX_5:
        ld      e_pointer,#14
        br      find_fuzz_row

;***************************[ Error_Negative]************************;
;********************************************************************;

ERROR_NEG:
        cmp     error,#-170
        jle     index1_170
        br      mid_neg

INDEX1_170:
        ld      e_pointer,#0
        br      find_fuzz_row

MID_NEG:
        cmp     error,#-85
        jlt     low_neg

HIGH_NEG:
        cmp     error,#-36
        jlt     high_neg_2

HIGH_NEG_1:
```

```
        cmp     error,#-5
        jge     index1_5
        ld      e_pointer,#10
        br      find_fuzz_row

INDEX1_5:
        ld      e_pointer,#12
        br      find_fuzz_row

HIGH_NEG_2:
        cmp     error,#-57
        jge     index1_57
        ld      e_pointer,#8
        br      find_fuzz_row

INDEX1_57:
        ld      e_pointer,#10
        br      find_fuzz_row

LOW_NEG:
        cmp     error,#-126
        jlt     low_neg_2

LOW_NEG_1:
        cmp     error,#-113
        jge     index1_113
        ld      e_pointer,#4
        br      find_fuzz_row

INDEX1_113:
        ld      e_pointer,#6
        br      find_fuzz_row

LOW_NEG_2:
        cmp     error,#-140
        jge     index1_140
        ld      e_pointer,#2
        br      find_fuzz_row

INDEX1_140:
        ld      e_pointer,#4
        br      find_fuzz_row


;******************[ Define Start Address of a Row ]*****************;
;********************************************************************;

FIND_FUZZ_ROW:

        bbs     deror,15,deror_neg

;**************************[ DError_Positive ]***********************;
;********************************************************************;

DEROR_POS:
```

```
        cmp     deror,#170              ;define 1st column in Fuzzy
matrix
        jge     de_index_170
        br      de_mid_pos

DE_INDEX_170:

        ld      de_pointer,#312
        br      fuzzy_result

DE_MID_POS:
        cmp     deror,#85
        jlt     de_low_pos

DE_HIGH_POS:
        cmp     deror,#126
        jlt     de_high_pos_2

DE_HIGH_POS_1:
        cmp     deror,#140
        jge     de_index_140
        ld      de_pointer,#260
        br      fuzzy_result

DE_INDEX_140:
        ld      de_pointer,#286
        br      fuzzy_result

DE_HIGH_POS_2:
        cmp     deror,#113
        jge     de_index_113
        ld      de_pointer,#234
        br      fuzzy_result

DE_INDEX_113:
        ld      de_pointer,#260
        br      fuzzy_result

DE_LOW_POS:
        cmp     deror,#43
        jlt     de_low_pos_2

DE_LOW_POS_1:
        cmp     deror,#57
        jge     de_index_57
        ld      de_pointer,#182
        br      fuzzy_result

DE_INDEX_57:
        ld      de_pointer,#208
        br      fuzzy_result

DE_LOW_POS_2:
        cmp     deror,#29
        jge     de_index_29
        ld      de_pointer,#156
        br      fuzzy_result
```

```
DE_INDEX_29:
        ld      de_pointer,#182
        br      fuzzy_result

;*************************[ DError_Negative ]***********************;
;*******************************************************************;

DEROR_NEG:
        cmp     deror,#-170
        jle     de_index1_170
        br      de_mid_neg

DE_INDEX1_170:
        ld      de_pointer,#0
        br      fuzzy_result

DE_MID_NEG:
        cmp     deror,#-85
        jlt     de_low_neg

DE_HIGH_NEG:
        cmp     deror,#-43
        jlt     de_high_neg_2

DE_HIGH_NEG_1:
        cmp     deror,#-29
        jge     de_index1_29
        ld      de_pointer,#130
        br      fuzzy_result

DE_INDEX1_29:
        ld      de_pointer,#156
        br      fuzzy_result

DE_HIGH_NEG_2:
        cmp     deror,#-57
        jge     de_index1_57
        ld      de_pointer,#104
        br      fuzzy_result

DE_INDEX1_57:
        ld      de_pointer,#130
        br      fuzzy_result

DE_LOW_NEG:
        cmp     deror,#-126
        jlt     de_low_neg_2

DE_LOW_NEG_1:
        cmp     deror,#-113
        jge     de_index1_113
        ld      de_pointer,#52
        br      fuzzy_result

DE_INDEX1_113:
        ld      de_pointer,#78
```

```
        br      fuzzy_result

DE_LOW_NEG_2:
        cmp     deror,#-140
        jge     de_index1_140
        ld      de_pointer,#26
        br      fuzzy_result

DE_INDEX1_140:
        ld      de_pointer,#52

;**********************-----------------------------*****************;
;**********************| End Fuzzy Logic Controller |*****************;
;**********************-----------------------------*****************;

FUZZY_RESULT:

        add     index,e_pointer,de_pointer
        ld      result,fuzzy[index]          ; start in Fuzzy matrix at
                                             ;511Eh


;**************************[ Fuzzy Integration ]**********************;
;*********************************************************************;

FUZZY_INT:

        ld      var_hi,#00
        ld      var_lo,result
        mul     var,#150
        add     U,var_lo,U_F_old            ;Wslip(K)=U_S*Ts+Wslip(K-1)

        bbs     U,15,pos_u
        br      limit_err_1

POS_U:
        neg     U
        inc     U


LIMIT_ERR_1:

        ld      U_F_old,U
        cmp     error,#5
        jle     Limit_err_2
        br      find_U_F

LIMIT_ERR_2:

        cmp     error,#-5
        jle     find_U_F
        ;br      start_calc
        ld      error,#0
        ld      old_error,#0
```

```
FIND_U_F:

        ld      var_hi,#00
        ld      var_lo,U
        mul     var,error
        shrl    var,#10
        ld      U_F,var_lo

;*****************************[ Integrator ]*************************;
;********************************************************************;

        cmp     error,#30
        jle     limit_low
        ld      U_int,#0
        br      find_wslip

LIMIT_LOW:
        cmp     error,#-30
        jge     integrate
        ld      U_int,#0
        br      find_wslip

INTEGRATE:

        ld      var_hi,#00
        ld      var_lo,error
        shll    var,#1
        add     U_int,var_lo,U_int_old

        ld      U_int_old,U_int
        cmp     U_int,#2240
        jge     U_int_pos                   ;-35 <= Wslip(rad/sec) <= 35
        cmp     U_int,#-2240
        jle     U_int_neg
        br      find_wslip

U_INT_POS:
        ld      U_int,#2240
        br      find_wslip

U_INT_NEG:
        ld      U_int,#-2240


;*****************************[ Find Wslip ]*************************;
;********************************************************************;

FIND_WSLIP:

        add     wslip,U_int,U_F

        cmp     wslip,#2240
        jge     limit_slip_pos                  ;-35 <= Wslip(rad/sec)
<= 35
        cmp     wslip,#-2240
        jle     limit_slip_neg
        br      find_Fm
```

```
LIMIT_SLIP_POS:
        ld      wslip,#2240
        br      find_Fm

LIMIT_SLIP_NEG:
        ld      wslip,#-2240

;*****************************[ Find Fm ]*************************;
;*****************************************************************;

FIND_FM:

        ld      var_hi,#00
        ld      var_lo,speed_last          ;var=Rot_freq
        shll    var,#2
        div     var,#19
        shll    var,#6
        ld      var2_hi,var_hi
        ld      var2_lo,var_lo
        add     var2_lo,Wslip
                                           ;var2=(Rot_freq+Wslip)*2^6
        shrl    var2,#6
        ld      Fm,var2_lo                 ;Fm=var2/2^6


;***********************[ PWM patterns Calculations ]*****************;
;*****************************************************************;

START_CALC:

        bbs     Tc_flg,0,start_calc1
        cmp     Fm,#314                  ;boost voltage for lower
frequencies
        jge     max_voltage              ;branch to boost
        br      boost                    ;else branch to find period

MAX_VOLTAGE:

        ld      Mr,#1000
        ld      Fm,#314
        br      find_per

BOOST:

        cmp     Fm,#62
        jle     clamp_boost          ;clamp boost for lower frequencies
        ld      var_hi,#0
        mulu    var,Fm,#815
        shrl    var,#8
        ld      Mr,var_lo
        br      find_per


CLAMP_BOOST:

        ld      Mr,#202                 ;min mod_ratio=0
```

```
        cmp     Fm,#6                   ;Fm=1 to avoid over_flow
        jle     clamp_low
        br      find_per

CLAMP_LOW:
        ld      Fm,#6

FIND_PER:

        ld      var_hi,#00
        mulu    var,Fm,#60
        ld      conv_const,var_lo       ;conv_const= Fm(rad) * 60

        ld      var_hi,#00
        ld      var_lo,Tc               ;var= Tc*(1024)^2
        shll    var,#10
        shll    var,#10
        divu    var,conv_const
        ld      Tc_sec,var_lo      ;Tc(sec)=Tc(deg)*(1024)^2/conv_const
        ld      var_hi,#00
        ld      time0,#0

        ldb     Tc_flg,#01b


START_CALC1:

        ld      var_hi,#00
        mulu    var,k,Tc_sec                    ; k*Tc in microseconds
        ld      angle_sec1,var_lo
        ld      var_hi,#00
        mulu    var,Tc,k                        ; K*Tc in degrees
        ld      angle_deg1,var_lo
        add     angle_deg2,angle_deg1,Tc2_deg   ; (K*Tc + Tc/2) in
degrees

        shl     angle_deg1,#1                  ; shift two addresess in
        shl     angle_deg2,#1                  ; sin table memory space


CONTIN_CALC:

        ld      var_lo,sin[angle_deg1]
        ld      var2_lo,sin[angle_deg2]
        ld      var_hi,#00
        ld      var2_hi,#00

        mulu    var,Mr                          ; Mr * sin(K*Tc)
        mulu    var2,Mr                         ; Mr * sin(K*Tc +Tc/2)
        shrl    var,#5                          ; divid over 32768
        shrl    var2,#5

        sub     ton,const,var_lo                ; 1 - Mr * sin(K*Tc)
        add     toff,var2_lo,#0                 ; Mr * sin(K*Tc+Tc/2)

        ld      ton_hi,#00
        ld      toff_hi,#00
```

```
        mulu    ton,Tc_sec                  ; Tc/4*(1-Mr*sin(K*Tc))
        mulu    toff,Tc_sec                 ; Tc/4 * Mr*sin(K*Tc+Tc/2))
        shrl    ton,#9
        shrl    toff,#9
        shrl    ton,#8
        shrl    toff,#8
        ld      var_hi,#00
        mulu    var,const1,Tc_sec              ; 3 * Tc/4
        shrl    var,#9
        shrl    var,#6
        add     toff_lo,var_lo                 ; Tc/4*(3+Mr*sin(K*Tc))

        add     ton_lo,angle_sec1     ; Tc*K+Tc/4*(1-Mr*sin(K*Tc))
        add     toff_lo,angle_sec1    ;Tc*K+Tc/4*(3+Mr*sin(K*Tc+Tc/2))

        sub     pulse_on,toff_lo,ton_lo  ;pulse_on= toff(k) - ton(k)
        sub     pulse_off,ton_lo,time0   ;pulse_off= ton(k) - toff(k-1)
        ld      time0,toff_lo


STORE_RESULT:

        st      pulse_off,[table1]+  ;update table with new caclulated
        st      pulse_on,[table1]+   ;switching time
        inc     k                    ;increment k
        cmp     k,Fr                 ;compare k with Fr/2 (half period)
        jge     clear_values         ;reset values if done
        br      start_calc1          ;else keep calculating


CLEAR_VALUES:

        ldb     start_flg,#0b        ;clear start flag
        ldb     start_soft,#0b
        ld      k,#0                 ;clear for new calculations
        ldb     Tc_flg,#00b
        ld      time0,#0


SWAP_TABLE1:

        jbs     swap_flg,0,swap_table2

        ld      table1,#3000h     ;first time table1 at 3000h is active
        ld      table2,#3400h     ;so store result in table2
        ldb     swap_flg,#01b     ;table1 is inactive at 3200h
        br      reset_pointers


SWAP_TABLE2:

        ld      table1,#3400h      ;table1 at 3200h for calculation
        ld      table2,#3000h      ;table2 at 3000h for PWM generation
        ldb     swap_flg,#00b
```

```
RESET_POINTERS:

        bbs     test1_flg,0,force_int
        br      loop


FORCE_INT:

        ld      time_ref1,timer1        ;update time reference for hso.1
        add     time_ref1,#2            ;provide synchronization
        ld      time_ref2,time_ref1     ;update time reference for hso.0
        ld      time_ref3,time_ref1     ;update time reference for hso.2

        add     table3,table2,#0         ;table3 = 0 deg to 60 deg
        add     table4,table2,pointer1   ;table4 = 60 deg to 120 deg
        add     table5,table2,pointer2   ;table5 = 120 deg to 180 deg

        orb     int_pending,#00001000b
        clrb    test1_flg



LOOP:

        br      detect_speed


;*********************[ Speed detection subroutine ]******************;
;*************************************************************************;

HSI_INT:

        pushf

        bbs     first_flg,0,first
        ld      per_st,timer1
        sub     period,per_st,per_end   ;define the detection period


FIRST_TIME:

        bbc     sw_set_flg,0,gen_first


FIRST:

        ldb     first_flg,#01b
        bbs     sw_exp_flg,0,update_speed

        popf
        ret


UPDATE_SPEED:

        ld      number,timer2
        ld      num2,period
```

```
        ldb     hsi_flg,#01b
        clrb    sw_exp_flg
        br      enable_swt


GEN_FIRST:

        ldb     sw_set_flg,#01b


ENABLE_SWT:

        ldb     ioc0,#10000110b


CHEK_FREE:

        bbs     ios0,7,chek_free
        ldb     hso_command,#00011000b
        add     hso_time,per_st,Ts
        ldb     sw_set_flg,#01b

        popf
        ret

;*********************************************************************;

SWT_INT:

        pushf

        andb    ioport1,#11111110b
        ld      per_end,per_st
        ldb     sw_exp_flg,#01b
        clrb    first_flg

        popf
        ret


;************************[ Update Mod_ratio ]*************************;
;*********************************************************************;

AD_INT:

        pushf

WAIT:

        jbs     ad_result_lo,3,wait

        andb    al,ad_result_lo,#11000000b
        ldb     ah,ad_result_hi
        shr     ax,#06h

        cmp     ax,#700
        jge     clamp
```

```
        andb    soft_flg,#10b
        ldb     compl_flg,#01b
        br      out_ad

CLAMP:

        orb     soft_flg,#01b

OUT_AD:

        ldb     start_flg,#01b
        ldb     start_soft,#01b

OUT1_AD:

        popf
        ret

;********************[ PWM Waveforms Generation ]*********************;
;********************[    1/6th of the cycle    ]*********************;

HSO_INT:

        pushf

        bbs     per1_flg,0,second_per     ;start switching first period
        inc     counter1

FIRST_PER2_OFF:

        jbs     ios0,7,first_per2_off
        add     time_ref2,[table4]

        ldb     hso_command,#00000010b       ;switch hso.2 off, T3
        ld      hso_time,time_ref2           ;update time reference
        add     table4,#2                    ;point to the next address

FIRST_PER1_ON:

        jbs     ios0,7,first_per1_on
        add     time_ref1,[table3]

        ldb     hso_command,#00100001b       ;switch hso.1 on, T1
        ld      hso_time,time_ref1           ;update time reference
        add     table3,#2                    ;point to the next address

FIRST_PER3_ON:

        jbs     ios0,7,first_per3_on
        add     time_ref3,[table5]
```

```
        ldb     hso_command,#00110011b      ;switch hso.3 on, T5
        ld      hso_time,time_ref3          ;update time reference
        add     table5,#2                   ;point to the next address

FIRST_PER2_ON:

        jbs     ios0,7,first_per2_on
        add     time_ref2,[table4]

        ldb     hso_command,#00100010b          ;switch hso.2 on, T3
        ld      hso_time,time_ref2
        add     table4,#2


FIRST_PER1_OFF:

        jbs     ios0,7,first_per1_OFF
        add     time_ref1,[table3]

        ldb     hso_command,#00000001b          ;switch hso.1 off, T1
        ld      hso_time,time_ref1
        add     table3,#2


FIRST_PER3_OFF:

        jbs     ios0,7,first_per3_OFF
        add     time_ref3,[table5]

        ldb     hso_command,#00000011b          ;switch hso.3 off, T5
        ld      hso_time,time_ref3
        add     table5,#2

        cmp     counter1,P_third                ;from 0 deg to 60 deg
        jge     clear1
        br      out1


clear1:
        clr     counter1
        ldb     per1_flg,#01b               ;first period has finished
        add     table3,table2,#0            ;update table3
        add     table4,table2,pointer1      ;update table4
        add     table5,table2,pointer2      ;update table5

        br      out1


;*************************************************************************:
;************************[ 2/6th of the cycle ]************************;


SECOND_PER:

        bbs     per1_flg,1,third_per        ;start switching 2nd period
        inc     counter1
```

```
SECOND_PER1_ON:

        jbs     ios0,7,second_per1_on
        add     time_ref1,[table4]

        ldb     hso_command,#00110001b      ;switch hso.1 on, T1
        ld      hso_time,time_ref1          ;update time reference
        add     table4,#2                   ;point to the next address


SECOND_PER3_OFF:

        jbs     ios0,7,second_per3_off
        add     time_ref3,[table3]

        ldb     hso_command,#00000011b      ;switch hso.3 off,T5
        ld      hso_time,time_ref3          ;update time reference
        add     table3,#2                   ;point to the next address


SECOND_PER2_OFF:

        jbs     ios0,7,second_per2_off
        add     time_ref2,[table5]

        ldb     hso_command,#00000010b      ;switch hso.2 off, T3
        ld      hso_time,time_ref2          ;update time reference
        add     table5,#2                   ;point to the next address


SECOND_PER1_OFF:

        jbs     ios0,7,second_per1_off
        add     time_ref1,[table4]

        ldb     hso_command,#00000001b        ;switch hso.1 off, T1
        ld      hso_time,time_ref1
        add     table4,#2


SECOND_PER3_ON:

        jbs     ios0,7,second_per3_on
        add     time_ref3,[table3]

        ldb     hso_command,#00100011b        ;switch hso.3 on, T5
        ld      hso_time,time_ref3
        add     table3,#2


SECOND_PER2_ON:

        jbs     ios0,7,second_per2_on
        add     time_ref2,[table5]
```

```
        ldb     hso_command,#00100010b          ;switch hso.0 on, T3
        ld      hso_time,time_ref2
        add     table5,#2

        cmp     counter1,P_third                ;from 60 deg to 120 deg
        jge     clear2
        br      out1


CLEAR2:

        clr     counter1
        ldb     per1_flg,#11b               ;second period has finished
        add     table3,table2,#0            ;update table3
        add     table4,table2,pointer1      ;update table4
        add     table5,table2,pointer2      ;update table5

        br       out1

;*************************************************************************;
;************************[ 3/6th of the cycle ]***********************;

THIRD_PER:

        bbs     per1_flg,2,fourth_per       ;start switching 3rd period
        inc     counter1


THIRD_PER2_OFF:

        jbs     ios0,7,third_per2_off
        add     time_ref3,[table4]

        ldb     hso_command,#00000011b       ;switch hso.3 off, T5
        ld      hso_time,time_ref3           ;update time reference
        add     table4,#2                    ;point to the next address


THIRD_PER3_ON:

        jbs     ios0,7,third_per3_on
        add     time_ref2,[table3]

        ldb     hso_command,#00100010b       ;switch hso.2 on, T3
        ld      hso_time,time_ref2           ;update time reference
        add     table3,#2                    ;point to the next address


THIRD_PER1_ON:

        jbs     ios0,7,third_per1_on
        add     time_ref1,[table5]

        ldb     hso_command,#00110001b       ;switch hso.1 on, T1
        ld      hso_time,time_ref1           ;update time reference
        add     table5,#2                    ;point to the next address
```

```
THIRD_PER2_ON:

        jbs     ios0,7,third_per2_on
        add     time_ref3,[table4]

        ldb     hso_command,#00100011b          ;switch hso.3 off, T5
        ld      hso_time,time_ref3
        add     table4,#2


THIRD_PER3_OFF:

        jbs     ios0,7,third_per3_off
        add     time_ref2,[table3]

        ldb     hso_command,#00000010b          ;switch hso.2 off, T3
        ld      hso_time,time_ref2
        add     table3,#2


THIRD_PER1_OFF:

        jbs     ios0,7,third_per1_off
        add     time_ref1,[table5]

        ldb     hso_command,#00000001b          ;switch hso.1 off, T1
        ld      hso_time,time_ref1
        add     table5,#2

        cmp     counter1,P_third                ;from 120 deg to 180 deg
        jge     clear3
        br      out1


CLEAR3:

        clr     counter1
        ldb     per1_flg,#111b                  ;Third period has finished
        add     table3,table2,#0                ;update table3
        add     table4,table2,pointer1          ;update table4
        add     table5,table2,pointer2          ;update table5

        br      out1

;*************************************************************************;
;************************[ 4/6th of the cycle ]***************************;

FOURTH_PER:

        bbs     per1_flg,3,fifth_per        ;start switching 4th period
        inc     counter1
```

```
FOURTH_PER2_ON:

        jbs     ios0,7,fourth_per2_on
        add     time_ref2,[table4]

        ldb     hso_command,#00110010b      ;switch hso.2 on, T3
        ld      hso_time,time_ref2          ;update time reference
        add     table4,#2                   ;point to the next address


FOURTH_PER3_OFF:

        jbs     ios0,7,fourth_per3_off
        add     time_ref1,[table3]

        ldb     hso_command,#00000001b      ;switch hso.1 off, T1
        ld      hso_time,time_ref1          ;update time reference
        add     table3,#2                   ;point to the next address


FOURTH_PER1_OFF:

        jbs     ios0,7,fourth_per1_off
        add     time_ref3,[table5]

        ldb     hso_command,#00000011b      ;switch hso.3 off, T5
        ld      hso_time,time_ref3          ;update time reference
        add     table5,#2                   ;point to the next address


FOURTH_PER2_OFF:

        jbs     ios0,7,fourth_per2_off
        add     time_ref2,[table4]

        ldb     hso_command,#00000010b          ;switch hso.2 off, T3
        ld      hso_time,time_ref2
        add     table4,#2


FOURTH_PER3_ON:

        jbs     ios0,7,fourth_per3_on
        add     time_ref1,[table3]

        ldb     hso_command,#00100001b          ;switch hso.1 on, T1
        ld      hso_time,time_ref1
        add     table3,#2


FOURTH_PER1_ON:

        jbs     ios0,7,fourth_per1_on
        add     time_ref3,[table5]

        ldb     hso_command,#00100011b          ;switch hso.3 off, T5
```

```
        ld      hso_time,time_ref3
        add     table5,#2

        cmp     counter1,P_third            ;from 180 deg to 240 deg
        jge     clear4
        br      out1


CLEAR4:

        clr     counter1
        ldb     per1_flg,#1111b             ;fourth period has finished
        add     table3,table2,#0            ;update table3
        add     table4,table2,pointer1      ;update table4
        add     table5,table2,pointer2      ;update table5

        br      out1


;*************************************************************************;
;************************[ 5/6th of the cycle ]***************************;

FIFTH_PER:

        bbs     per1_flg,4,sixth_per        ;start switching 5th period
        inc     counter1


FIFTH_PER2_OFF:

        jbs     ios0,7,fifth_per2_off
        add     time_ref1,[table4]

        ldb     hso_command,#00000001b      ;switch hso.1 off, T1
        ld      hso_time,time_ref1          ;update time reference
        add     table4,#2                   ;point to the next address


FIFTH_PER3_ON:

        jbs     ios0,7,fifth_per3_on
        add     time_ref3,[table3]

        ldb     hso_command,#00110011b      ;switch hso.3 on, T5
        ld      hso_time,time_ref3          ;update time reference
        add     table3,#2                   ;point to the next address


FIFTH_PER1_ON:

        jbs     ios0,7,fifth_per1_on
        add     time_ref2,[table5]

        ldb     hso_command,#00100010b      ;switch hso.2 on, T3
        ld      hso_time,time_ref2          ;update time reference
        add     table5,#2                   ;point to the next address
```

```
FIFTH_PER2_ON:

        jbs     ios0,7,fifth_per2_on
        add     time_ref1,[table4]

        ldb     hso_command,#00100001b          ;switch hso.1 on, T1
        ld      hso_time,time_ref1
        add     table4,#2


FIFTH_PER3_OFF:

        jbs     ios0,7,fifth_per3_off
        add     time_ref3,[table3]

        ldb     hso_command,#00000011b          ;switch hso.3 off, T5
        ld      hso_time,time_ref3
        add     table3,#2


FIFTH_PER1_OFF:

        jbs     ios0,7,fifth_per1_off
        add     time_ref2,[table5]

        ldb     hso_command,#00000010b        ;switch hso.2 off, T3
        ld      hso_time,time_ref2
        add     table5,#2

        cmp     counter1,P_third              ;from 240 deg to 300 deg
        jge     clear5
        br      out1


CLEAR5:

        clr     counter1
        ldb     per1_flg,#11111b            ;fifth period has finished
        add     table3,table2,#0            ;update table3
        add     table4,table2,pointer1      ;update table4
        add     table5,table2,pointer2      ;update table5

        br      out1

;*********************************************************************;
;***********************[ 6/6th of the cycle ]************************;

SIXTH_PER:

        inc     counter1


SIXTH_PER2_ON:

        jbs     ios0,7,sixth_per2_on
        add     time_ref3,[table4]
```

```
        ldb     hso_command,#00110011b      ;switch hso.3 on, T5
        ld      hso_time,time_ref3          ;update time reference
        add     table4,#2                   ;point to the next address

SIXTH_PER3_OFF:

        jbs     ios0,7,sixth_per3_off
        add     time_ref2,[table3]

        ldb     hso_command,#00000010b      ;switch hso.2 off, T3
        ld      hso_time,time_ref2          ;update time reference
        add     table3,#2                   ;point to the next address

SIXTH_PER1_OFF:

        jbs     ios0,7,sixth_per1_off
        add     time_ref1,[table5]

        ldb     hso_command,#00000001b      ;switch hso.1 off, T1
        ld      hso_time,time_ref1          ;update time reference
        add     table5,#2                   ;point to the next address

SIXTH_PER2_OFF:

        jbs     ios0,7,sixth_per2_off
        add     time_ref3,[table4]

        ldb     hso_command,#00000011b      ;switch hso.3 off, T5
        ld      hso_time,time_ref3
        add     table4,#2

SIXTH_PER3_ON:

        jbs     ios0,7,sixth_per3_on
        add     time_ref2,[table3]

        ldb     hso_command,#00100010b      ;switch hso.2 on, T3
        ld      hso_time,time_ref2
        add     table3,#2

SIXTH_PER1_ON:

        jbs     ios0,7,sixth_per1_on
        add     time_ref1,[table5]

        ldb     hso_command,#00100001b    ;switch hso.1 on, T1
        ld      hso_time,time_ref1
        add     table5,#2
```

```
        cmp     counter1,P_third          ;Start switching last period
        jge     clear                     ;from 300 deg to 360 deg
        br      out1


CLEAR:

        clr     counter1
        ldb     per1_flg,#0b              ;last period has finished

;***************************[ Update Tables ]*************************;

SWAP_TAB1:

        jbc     swap_flg,0,swap_tab2

        ld      table1,#3000h
        ld      table2,#3400h
        ldb     swap_flg,#01b

        add     table3,table2,#0
        add     table4,table2,pointer1
        add     table5,table2,pointer2
        br      out


SWAP_TAB2:

        ld      table1,#3400h
        ld      table2,#3000h
        ldb     swap_flg,#00b

        add     table3,table2,#0
        add     table4,table2,pointer1
        add     table5,table2,pointer2


out:
        ldb     ad_command,#00001100b    ;start conversion every period
        nop
        nop


OUT1:

        popf
        ret

;************************[ Not used Vectors ]*************************;

not_used:

        ret
```

```
;******************[ Fuzzy Logic Look-Up Table ]**********************;

CSEG at 5000h

Fuzzy:   ;Fuzzy Logic lookup table result

        dcw      0 , 19, 38, 38, 38, 47, 59, 59, 59, 63, 77, 78, 78
        dcw     -19, 0 , 19, 19, 38, 47, 47, 48, 59, 63, 63, 63, 78
        dcw     -38,-19, 0 , 19, 38, 38, 38, 48, 59, 59, 59, 63, 78
        dcw     -38,-19,-19, 0 , 19, 19, 38, 47, 47, 47, 59, 63, 63
        dcw     -38,-38,-38,-19, 0 , 19, 38, 38, 38, 47, 59, 59, 59
        dcw     -48,-48,-37,-19,-19, 0 , 19, 19, 38, 47, 48, 48, 59
        dcw     -59,-48,-37,-37,-37,-19, 0 , 19, 38, 38, 38, 47, 59
        dcw     -59,-48,-48,-48,-37,-19,-19, 0 , 19, 19, 38, 47, 47
        dcw     -59,-59,-59,-48,-37,-37,-37,-19, 0 , 19, 38, 38, 38
        dcw     -63,-63,-59,-48,-48,-48,-37,-19,-19, 0 , 19, 19, 38
        dcw     -78,-63,-59,-59,-59,-48,-37,-37,-37,-19, 0 , 19, 38
        dcw     -78,-63,-63,-63,-59,-48,-46,-46,-37,-19,-19, 0 , 19
        dcw     -78,-78,-78,-63,-59,-59,-59,-46,-37,-37,-37,-19, 0



;************************[ SIN Look-Up Table ]************************;


CSEG at 5500h

sin:     ;sin table to be loaded in the equation.

        dcw  0,31,62,94,125,156,186,217,247,276,306,335,363,391,419,446
        dcw  472,498,524,548,572,595,617,639,660,680,699,718,735,752
        dcw  768,783,797,811,823,835,845,855,864,873,880,887,892,897
        dcw  902,905,908,910,911,912,912,912,911,910,908,905,902,899
        dcw  895,891,887,882,877,872,867,862,856,851,845,840,834,829
        dcw  823,818,813,808,803,799,795,790,787,783,780,780,777,775
        dcw  773,771,770,769,768,768,768,769,770,771,773,775,777,780
        dcw  783,787,790,795,799,803,808,813,818,823,829,834,840,845
        dcw  851,856,862,867,872,877,882,887,891,895,899,902,905,908
        dcw  910,911,912,912,912,911,910,908,905,902,897,892,887,880
        dcw  873,864,855,845,835,823,811,797,783,768,752,735,718,699
        dcw  680,660,639,617,595,572,548,524,498,472,446,419,391,363
        dcw  335,306,276,247,217,186,156,125,94,62,31,0,-31,-62-94


end
```